Corso di Algoritmi e Strutture Dati

TEMI D'ESAME

Anni accademici dal 2002/2003 al 2009/2010.

Corso di laurea in Informatica
Università degli Studi di Milano

Settembre 2010

Docente del corso: Massimiliano Goldwurm

ALGORITMI E STRUTTURE DATI

Prova scritta del 3.9.2010

Esercizio 1.

a) Dare la definizione di "partizione" di un insieme. Come possiamo rappresentare una partizione mediante una relazione di equivalenza?

b) Definire le operazioni UNION-FIND su partizioni.

Esercizio 2.

Considera la seguente procedura che, a partire dalla partizione identità , esegue una sequenza di operazioni UNION-FIND sull'insieme degli interi $\{1, 2, \ldots, n\}$, dove $n > 2$.

```
begin
      for i = 2,...,n do
          if i ≡ 1 (mod 3)  then UNION(i,1)
                            else UNION(1,i)
end
```

a) Supponendo di utilizzare una foresta semplice, descrivere l'albero ottenuto dalla procedura data per un intero $n > 2$ qualsiasi.

b) Nell'ipotesi precedente, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

c) Supponiamo ora di utilizzare una foresta con bilanciamento: descrivere l'albero ottenuto eseguendo la procedura e valutare il tempo di calcolo richiesto in questo caso.

Esercizio 3.

Dato un albero con radice, chiamiamo *peso* di un nodo v il numero razionale $P(v)$ così definito:

$$P(v) = \begin{cases} 1 & \text{se } v \text{ è una foglia,} \\ k + \frac{P(w_1)+P(w_2)+\cdots+P(w_k)}{k} & \text{se } v \text{ è un nodo interno e} \\ & w_1, w_2, \ldots, w_k \text{ sono i suoi figli.} \end{cases}$$

a) Descrivere una procedura che, avendo in input un albero con radice (rappresentato mediante liste di adiacenza), calcoli il *peso* della radice.

b) Assumendo il criterio di costo uniforme, valutare il tempo di calcolo e lo spazio di memoria richiesti in funzione del numero n di nodi dell'albero.

ALGORITMI E STRUTTURE DATI
Prova scritta del 1.7.2010

Esercizio 1.
Considera le sequenza $\{g_n\}_{n\geq 1}$ così definita:

$$g_n = \begin{cases} 5n^3 + 6n^2 & \text{se } n \text{ è pari} \\ 3n^2\lfloor \log n \rfloor + 4n & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare quelle vere e quelle false:

$$1)\ g_n = o(n^3), \quad 2)\ g_n = O(n^3), \quad 3)\ g_n = \Theta(n^3), \quad 4)\ g_n = O(n^4).$$

Esercizio 2.
a) Definire la nozione di Heap.
b) Eseguire l'algoritmo Heapsort sull'input

$$3, 4, 5, 6, 1, 0, 7, 9, 2, 8$$

mettendo in evidenza gli scambi compiuti.

Esercizio 3.
a) Definire la nozione di grafo orientato e quello di distanza tra due nodi in un tale grafo.
b) Descrivere una procedura per risolvere il seguente problema:

Istanza : un grafo orientato G di n nodi e m lati, rappresentato mediante liste di adiacenza e un nodo s di G;
Soluzione : un nodo u in G che si trova a distanza massima da s.

c) Assumendo il criterio di costo uniforme, valutare in funzione di n e m il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo nel caso peggiore.

ALGORITMI E STRUTTURE DATI
Prova scritta del 15.6.2010

Esercizio 1. Considera le sequenza $\{g_n\}_{n\geq 1}$ così definita:

$$g_n = \begin{cases} 2n^3 + 5n & \text{se } n \text{ è pari} \\ 3n^2 + 4\lfloor \log n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare quelle vere e quelle false:

$$1)\ g_n = O(n^3), \quad 2)\ g_n = \Theta(n^3), \quad 3)\ g_n = o(n^3 \log n), \quad 4)\ g_n = O(n^2).$$

Esercizio 2. Eseguire l'algoritmo Quicksort sull'input

$$6, 9, 8, 4, 1, 7, 5, 2, 0, 3$$

assumendo sempre come pivot il primo elemento della sequenza corrente e mettendo in evidenza gli scambi compiuti.

Esercizio 3. Consideriamo l'algoritmo che su input $n \in \mathbb{N}$, $n > 2$, introduce in un albero di ricerca binaria inizialmente vuoto la seguente sequenza di numeri interi, utilizzando la tradizionale procedura di inserimento:

$$n, 2n, n-1, 2n-1, n-2, 2n-2, \ldots, 1, n+1$$

a) Descrivere l'albero ottenuto assumendo il naturale ordinamento sugli interi.
b) Valutare in funzione di n l'ordine di grandezza del tempo di calcolo richiesto dall'algoritmo.

Esercizio 4.
a) Definire la nozione di albero con radice e quella di altezza di un nodo in tale albero.
b) Descrivere una procedura per risolvere il seguente problema:

Istanza : un albero con radice T di n nodi, rappresentato mediante liste di adiacenza.
Soluzione : l'altezza di ciascun nodo di T.

c) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dalla procedura precedente in funzione del parametro n. Giustificare le soluzioni fornite.

ALGORITMI E STRUTTURE DATI

Prova scritta del 11.2.2010

Esercizio 1.

Eseguire l'algoritmo Mergesort sull'input

$$4, 3, 6, 5, 1, 3, 5, 0, 9, 8$$

mettendo in evidenza i confronti eseguiti.

Esercizio 2.

Considera il problema di calcolare l'espressione

$$\big(a_1 - a_2 + a_3 - a_4 + \cdots \ + (-1)^n a_{n-1} + (-1)^{n+1} a_n\big)$$

assumendo come input la sequenza di numeri interi $n, a_1, a_2, \ldots, a_n$ (dove $n \geq 1$).

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n. Giustificare le valutazioni fornite.

Esercizio 3.

Ricordiamo che in un grafo orientato la distanza di un nodo b da un nodo a è la lunghezza del più corto cammino da a a b.

a) Tenendo conto della definizione appena data, descrivere un algoritmo per il seguente problema:

> Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e un nodo $s \in V$.
>
> Soluzione : per ogni nodo $v \in V$ la distanza di v da s (con la convenzione di restituire il valore $\perp$ se v non è raggiungibile da s).

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

c) In che modo possiamo rappresentare efficientemente i cammini di lunghezza minima dal nodo s a ciascuna altro nodo del grafo? Come possiamo modificare l'algoritmo presentato nel punto a) per calcolare tale rappresentazione?

ALGORITMI E STRUTTURE DATI
Prova scritta del 28.01.2010

Esercizio 1.
Considera le sequenza $\{g_n\}_{n\geq 1}$ così definita:

$$g_n = \begin{cases} 6n + 2n^2 & \text{se } n \text{ è pari} \\ 5n\lfloor \log n \rfloor + 3n & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare quelle vere e quelle false, giustificando le risposte:

$$1)\ g_n = O(n^2), \quad 2)\ g_n = o(n^2), \quad 3)\ g_n = \Theta(n^2), \quad 4)\ g_n = O(n^3).$$

Esercizio 2.
Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND sull'insieme degli interi $\{1, 2, \ldots, n\}$, dove $n > 2$, a partire dalla partizione identità .

```
begin
      for i = 2,...,n do
          if i pari   then UNION(i,1)
                      else UNION(i-1,i)
end
```

a) Supponendo di utilizzare una foresta semplice, descrivere l'albero ottenuto dalla procedura data.

b) Nell'ipotesi precedente, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

c) Supponiamo ora di utilizzare una foresta con bilanciamento: descrivere l'albero ottenuto eseguendo la procedura e valutare il tempo di calcolo richiesto in questo caso.

Esercizio 3.
a) Descrivere una procedura per risolvere il seguente problema:

> Istanza : un albero con radice T di n nodi, rappresentato mediante liste di adiacenza.
> Soluzione : la profondità di ciascun nodo di T.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dalla procedura precedente in funzione del parametro n. Giustificare le soluzioni fornite.

ALGORITMI E STRUTTURE DATI
Prova scritta del 4.9.2009

Esercizio 1.
Eseguire l'algoritmo Mergesort sull'input

$$2, 7, 1, 0, 4, 5, 8, 3, 9, 6$$

mettendo in evidenza i confronti eseguiti.

Esercizio 2.
a) Definire la nozione di coda come struttura dati.
b) Descrivere ad alto livello un algorimo di esplorazione di grafi basato sull'uso di una coda.

Esercizio 3.
a) Definire la nozione di albero con radice e quella di altezza di un nodo in tale albero.
b) Descrivere una procedura per risolvere il seguente problema:

Istanza : un albero con radice T di n nodi, rappresentato mediante liste di adiacenza.
Soluzione : l'altezza di ciascun nodo di T.

c) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dalla procedura precedente in funzione del parametro n. Giustificare le soluzioni fornite.

ALGORITMI E STRUTTURE DATI
Prova scritta del 7.7.2009

Esercizio 1.

Determinare tra le sequenti relazioni quelle vere e quelle false:

a) $O(n) - n + 1 = O(1)$,

b) $(2 + (-1)^n)n^2 + 3n = o(n^2 \log n)$,

c) $n^2(2 + \cos n) - n = \Theta(n^2)$,

d) $O(n \log n) \cdot \frac{1}{\log n} = \Theta(n)$.

Esercizio 2.

Considera il problema di calcolare l'espressione

$$\left(3a_1 + a_2 + 3a_3 + \cdots\ + (2 + (-1)^{n+1})a_n\right)$$

assumendo come input la sequenza di numeri interi $n, a_1, a_2, \ldots, a_n$ (dove $n \geq 1$).

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n. Giustificare le valutazioni fornite.

Esercizio 3.

Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \ldots, a_n\}$, a partire dalla partizione identità .

```
begin
      for i = 2,...,n do
          if i pari   then UNION(a_1, a_i)
                      else UNION(a_i, a_{i-2})
end
```

a) Supponendo di utilizzare una foresta semplice, descrivere l'albero ottenuto dalla procedura data.

b) Nell'ipotesi precedente, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

c) Supponiamo ora di utilizzare una foresta con bilanciamento: descrivere l'albero ottenuto eseguendo la procedura e valutare il tempo di calcolo richiesto in questo caso.

ALGORITMI E STRUTTURE DATI
Prova scritta del 11.6.2009

Esercizio 1. Eseguire l'algoritmo Quicksort sull'input

$$7, 6, 5, 3, 2, 0, 8, 9, 4, 1$$

assumendo sempre come pivot il primo elemento della sequenza corrente e mettendo in evidenza gli scambi compiuti.

Esercizio 2. Consideriamo l'algoritmo che su input $n \in \mathbb{N}$, $n > 2$, introduce in un albero di ricerca binaria inizialmente vuoto la seguente sequenza di numeri interi, utilizzando la tradizionale procedura di inserimento:

$$n, 2n, n-1, 2n-1, n-2, 2n-2, \ldots, 1, n+1$$

a) Descrivere l'albero ottenuto assumendo il tradizionale ordinamento sugli interi.
b) Valutare in funzione di n l'ordine di grandezza del tempo di calcolo richiesto dall'algoritmo.

Esercizio 3.
a) Ricordando la procedura di visita in profondità dei grafi non orientati, descrivere un algoritmo per il seguente problema:

Istanza : un grafo non orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza.
Soluzione : il valore 1 se G possiede un ciclo, il valore 0 altrimenti.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

Esercizio 4.
a) Definire la struttura dati associata alle operazioni UNION, FIND.
b) Descrivere una implementazione di tale struttura.

ALGORITMI E STRUTTURE DATI
Prova scritta del 19.2.2009

Esercizio 1.
Considera le sequenza $\{g_n\}_{n\geq 1}$ così definita:

$$g_n = \begin{cases} 3n^3 + 2n & \text{se } n \text{ è pari} \\ n^2 + \lfloor \log n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare quelle vere e quelle false:

$$1)\ g_n = O(n^3), \quad 2)\ g_n = \Theta(n^3), \quad 3)\ g_n = o(n^3 \log n), \quad 4)\ g_n = O(n^2).$$

Esercizio 2. Eseguire l'algoritmo Heapsort sull'input

$$3, 4, 5, 6, 1, 0, 7, 9, 2, 8$$

mettendo in evidenza gli scambi compiuti.

Esercizio 3.
a) Definire la nozione di albero di ricerca binaria.
b) Usando la tradizionale procedura di inserimento, descrivere l'albero di ricerca binaria che si ottiene inserendo la seguente sequenza di interi a partire dall'albero vuoto:

$$7, 3, 5, 1, 4, 2, 6, 0, 10, 8, 12, 9, 11$$

c) Descrivere l'albero di ricerca binaria che si ricava cancellando i valori 3 e 10 dall'albero ottenuto nell'esercizio precedente.

Esercizio 4.
a) Descrivere un algoritmo del tipo *divide et impera* che su input $n, k, b_1, b_2, \ldots, b_n \in \mathbf{N}$ calcola il valore

$$F(n, k, \underline{b}) = b_1^k + b_2^k + \cdots + b_n^k \ .$$

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione dei parametri n e k.

ALGORITMI E STRUTTURE DATI
Prova scritta del 29.1.2009

Esercizio 1.
Considera le sequenza $\{g_n\}_{n\geq 1}$ così definita:

$$g_n = \begin{cases} 3n^2 + 4n & \text{se } n \text{ è pari} \\ n^3 + \lfloor \log n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare quelle vere e quelle false:

$$1)\ g_n = O(n^2), \quad 2)\ g_n = \Theta(n^3), \quad 3)\ g_n = o(n^3 \log n), \quad 4)\ g_n = O(n^3).$$

Esercizio 2. Eseguire l'algoritmo Mergesort sull'input

$$3, 4, 1, 5, 6, 0, 5, 3, 8, 9$$

mettendo in evidenza i confronti eseguiti.

Esercizio 3.
Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \ldots, a_n\}$, a partire dalla partizione identità , supponendo n pari.

```
begin
      for i = 2, ..., n do
          if i pari   then UNION(a_1, a_i)
                      else UNION(a_i, a_{i-1})
end
```

a) Supponendo di utilizzare una foresta semplice come struttura dati per l'esecuzione delle operazioni UNION-FIND, determinare l'albero ottenuto eseguendo la procedura descritta su input $n \in \mathbf{N}$ pari. Assumendo il criterio uniforme, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n .

b) Determinare l'albero ottenuto supponendo di utilizzare come struttura dati una foresta con bilanciamento e valutare il tempo di calcolo richiesto in questo caso assumendo il criterio uniforme.

Esercizio 4.
a) Definire la nozione di albero con radice. Definire inoltre l'altezza di un nodo in tale albero.

b) Descrivere una procedura per risolvere il seguente problema:

> Istanza : un albero con radice T di n nodi, rappresentato mediante liste di adiacenza.
> Soluzione : l'altezza di ciascun nodo di T.

c) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dalla procedura precedente in funzione del parametro n. Giustificare le soluzioni fornite.

ALGORITMI E STRUTTURE DATI
Prova scritta del 4.9.2008

Esercizio 1.
Eseguire l'algoritmo Mergesort sull'input

$$8, 3, 7, 0, 9, 5, 4, 6, 2, 1$$

mettendo in evidenza i confronti eseguiti.

Esercizio 2.
a) Definire la nozione di pila come struttura dati.
b) Definire le operazioni UNION-FIND.

Esercizio 3.
a) Descrivere l'albero di ricerca binaria che si ottiene inserendo la seguente sequenza di interi a partire dall'albero vuoto:

$$7, 3, 5, 1, 4, 2, 6, 0, 10, 8, 12, 9, 11$$

b) Descrivere l'albero di ricerca binaria che si ricava cancellando i valori 3 e 10 dall'albero ottenuto nell'esercizio precedente.

Esercizio 4.
a) Descrivere un algoritmo per il seguente problema:

> Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e un nodo $s \in V$.
>
> Soluzione : per ogni nodo $v \in V$ la distanza di v da s (con la convenzione di restituire il valore $\perp$ se v non è raggiungibile da s).

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 2.7.2008

Esercizio 1.
Considera le sequenza $\{g_n\}_{n\geq 1}$ così definita:

$$g_n = \begin{cases} 2n+1 & \text{se } n \text{ è pari} \\ n^2 + \lfloor \log n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ g_n = O(n^2), \quad 2)\ g_n = \Theta(n^2), \quad 3)\ 2g_n = O(n^2 \log n), \quad 4)\ g_n = o(n^2).$$

Esercizio 2. Eseguire l'algoritmo Quicksort sull'input $4, 9, 8, 0, 5, 1, 7, 2, 8, 3$ assumendo sempre come pivot il primo elemento della sequenza corrente e mettendo in evidenza gli scambi compiuti.

Esercizio 3.
Considera il problema di calcolare il valore dell'espressione

$$\sum_{i=1}^{n} |a_i - b_i|$$

assumendo come input la sequenza di numeri interi $n, a_1, b_1, a_2, b_2, \ldots, a_n, b_n$, dove $n \geq 1$.

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n. Giustificare le valutazioni fornite.

Esercizio 4. Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \ldots, a_n\}$, a partire dalla partizione identità (supponendo n pari).

```
begin
      for i = 1, ..., n − 1 do
          if i dispari   then UNION(a_i, a_{i+1})
                         else UNION(a_i, a_2)
end
```

a) Supponendo di utilizzare una foresta semplice, determinare la foresta ottenuta al termine della procedura descritta. Assumendo il criterio di costo uniforme, valutare inoltre l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n.

b) Svolgere l'esercizio precedente supponendo di utilizzare una foresta con bilanciamento.

ALGORITMI E STRUTTURE DATI
Prova scritta del 12.6.2008

Esercizio 1.
Eseguire l'algoritmo Mergesort sull'input

$$4, 5, 2, 6, 7, 1, 3, 0, 9, 8$$

mettendo in evidenza i confronti eseguiti.

Esercizio 2.
a) Definire la nozione di coda come struttura dati.
b) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e un nodo $s \in V$.
Soluzione : per ogni nodo $v \in V$ la distanza di v da s (con la convenzione di restituire il valore $\perp$ se v non è raggiungibile da s).

c) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

Esercizio 3.

a) Qual è l'altezza minima di un albero di ricerca binaria di n nodi?

b) Usando la tradizionale procedura di inserimento di un nodo in un albero di ricerca binaria, descrivere un algoritmo che su input $n \in \mathbb{N}$, $n > 0$, restituisce un albero di ricerca binaria per l'insieme $\{1, 2, \ldots, n\}$ di altezza minima.

c) Valutare il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo descritto nel punto precedente.

ALGORITMI E STRUTTURE DATI
Prova scritta del 19.2.2008

Esercizio 1.

Considera le sequenza $\{g_n\}$ così definita:

$$g_n = \begin{cases} 2n\lceil \log n \rceil + 5n & \text{se } n \text{ è pari} \\ 3n^3 + 3n^2 \lfloor \log n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ g_n = O(n^2 \log n), \quad 2)\ g_n = O(e^n), \quad 3)\ g_n = \Theta(n^3), \quad 4)\ g_n = o(n^3).$$

Esercizio 2.

Definire la nozione di riducibilità polinomiale tra problemi di decisione e spiegarne brevemente il significato intuitivo.

Esercizio 3.

Considera il problema di calcolare l'espressione

$$(a_1 + 3a_2 + a_3 + \cdots + (2 + (-1)^n)a_n)$$

assumendo come input la sequenza di numeri interi $n, a_1, a_2, \ldots, a_n$ (dove $n \geq 1$).

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n. Giustificare le valutazioni fornite.

Esercizio 4.

a) Descrivere una procedura per risolvere il seguente problema:

> Istanza : un albero con radice T di n nodi, rappresentato mediante liste di adiacenza.
> Soluzione : il numero di discendenti di ciascun nodo di T.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dalla procedura precedente in funzione del parametro n. Giustificare le soluzioni fornite.

ALGORITMI E STRUTTURE DATI
Prova scritta del 29.1.2008

Esercizio 1.

Determinare tra le sequenti relazioni quelle vere e quelle false:

a) $O(n) - n + 1 = O(1)$,

b) $(2 + (-1)^n)n^2 + 3n = o(n^2 \log n)$,

c) $n^2(2 + \cos n) - n = \Theta(n^2)$,

d) $O(n \log n) \cdot \frac{1}{\log n} = \Theta(n)$.

Esercizio 2.

Eseguire l'algoritmo Heapsort sull'input

$$3, 4, 5, 6, 1, 0, 7, 9, 2, 8$$

mettendo in evidenza gli scambi compiuti.

Esercizio 3.

Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \ldots, a_n\}$, a partire dalla partizione identità .

```
begin
      for i = 2,...,n do
          if i e' pari   then UNION(a_i, a_1)
                         else UNION(a_2, a_i)
end
```

a) Supponendo di utilizzare una foresta semplice, descrivere l'albero ottenuto dalla procedura data.

b) Nell'ipotesi precedente, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

c) Supponiamo ora di utilizzare una foresta con bilanciamento: descrivere l'albero ottenuto eseguendo la procedura e valutare il tempo di calcolo richiesto in questo caso.

Esercizio 4.

a) Descrivere una procedura per risolvere il seguente problema:

> Istanza : un grafo orientato G di n nodi e m lati, rappresentato mediante liste di adiacenza e un nodo s di G;
>
> Soluzione : un nodo u in G che si trova a distanza massima da s.

b) Assumendo il criterio di costo uniforme, valutare in funzione di n e m il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo nel caso peggiore.

ALGORITMI E STRUTTURE DATI
Prova scritta del 9.1.2008

Esercizio 1.

Determinare tra le sequenti relazioni quelle vere e quelle false:

a) $O(n) - n + 1 = O(1)$,

b) $\Theta(n) + 3n = \Theta(n)$,

c) $O(n^2) - n^2 = o(n^2)$,

d) $O(n \log n) \cdot \frac{1}{\log n} = O(n)$.

Esercizio 2.

Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \ldots, a_n\}$, a partire dalla partizione identità .

```
begin
      for i = 2, ..., n do
          if i pari   then UNION(a_1, a_i)
                      else UNION(a_i, a_{i-2})
end
```

a) Supponendo di utilizzare una foresta semplice, descrivere l'albero ottenuto dalla procedura data.

b) Nell'ipotesi precedente, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

c) Supponiamo ora di utilizzare una foresta con bilanciamento: descrivere l'albero ottenuto eseguendo la procedura e valutare il tempo di calcolo richiesto in questo caso.

Esercizio 3.

a) Qual è l'altezza minima di un albero di ricerca binaria di n nodi?

b) Usando la tradizionale procedura di inserimento di un nodo in un albero di ricerca binaria, descrivere un algoritmo che su input $n \in \mathbf{N}$, $n > 0$, restituisce un albero di ricerca binaria per l'insieme $\{1, 2, \ldots, n\}$ di altezza minima.

c) Valutare il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo descritto nel punto precedente.

ALGORITMI E STRUTTURE DATI
Prova scritta del 5.9.2007

Esercizio 1.

Dato un intero $n > 2$, sia T l'albero di ricerca binaria formato da un solo nodo di etichetta n. Eseguiamo su T il seguente algoritmo che introduce nuovi nodi nell'albero utilizzando la tradizionale procedura di inserimento negli alberi di ricerca binaria.

```
for  i = 1, ..., n − 1 do
    begin
          inserisci in T un nodo di etichetta i
          inserisci in T un nodo di etichetta n + i
    end
```

a) Descrivere l'albero di ricerca binaria prodotto dall'algoritmo.

b) Valutare in funzione di n l'ordine di grandezza del tempo di calcolo richiesto assumendo il criterio uniforme.

Esercizio 2.

Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \ldots, a_n\}$, a partire dalla partizione identità .

```
begin
     for i = 2, ..., n do
         if i pari   then UNION(a_1, a_i)
                     else UNION(a_i, a_{i−2})
end
```

a) Supponendo di utilizzare una foresta semplice, descrivere l'albero ottenuto dalla procedura data.

b) Nell'ipotesi precedente, valutare l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

c) Supponiamo ora di utilizzare una foresta con bilanciamento: descrivere l'albero ottenuto eseguendo la procedura e valutare il tempo di calcolo richiesto in questo caso.

Esercizio 3.

Ricordiamo che in un grafo orientato G un nodo v è *raggiungibile* da un nodo s se esiste in G un cammino da s a v.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E\rangle$ rappresentato mediante liste di adiacenza e due nodi $s, u \in V$.

Soluzione : la lunghezza del più corto cammino da s a u se u è raggiungibile da s, altrimenti il valore convenzionale $\perp$.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 12.6.2007

Esercizio 1.

Eseguire l'algoritmo Mergesort sull'input

$$3, 4, 1, 5, 6, 0, 5, 3, 8, 9$$

mettendo in evidenza i confronti eseguiti.

Esercizio 2.

a) Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ 16n \log n + 3 + 2^{-n} = o(n^2/\log n) \quad 2)\ 16n \log n + 3 + 2^{-n} = O(n^2)$$
$$3)\ 16n \log n + 3 + 2^{-n} = \Theta(n^2) \quad 4)\ 16n \log n + 3 + 2^{-n} = \Theta(n \log n)$$

b) Fornire un esempio di una sequenza $\{f_n\}_n \subseteq \mathbb{R}_+$ tale che $f_n = O(n^2)$, che **non** soddisfi le relazioni $f_n = \Theta(n^2)$ e $f_n = o(n^2)$.

Esercizio 3.

a) Definire la nozione di pila come struttura dati.

b) Descrivere un algorimo basato sulla gestione di una pila per la visita in preordine di un albero binario (rappresentato mediante la radice e i tradizionali vettori *sin* e *des*).

c) Assumendo il criterio di costo uniforme, valutare il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo descritto al punto precedente in funzione del numero n di nodi dell'albero.

Esercizio 4.

Ricordiamo che in un grafo orientato G un nodo v è *raggiungibile* da un nodo s se esiste in G un cammino da s a v.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e due nodi $s, u \in V$.

Soluzione : la lista dei nodi che formano un cammino di lunghezza minima da s a u se u è raggiungibile da s, altrimenti il valore convenzionale $\perp$.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 8.2.2007

Esercizio 1.

Eseguire l'algoritmo Heapsort sull'input

$$2, 1, 6, 5, 4, 7, 9, 8, 3$$

mettendo in evidenza gli scambi compiuti.

Esercizio 2.

a) Fornire un esempio di una sequenza $\{f_n\}_{n\in\mathbb{N}} \subseteq \mathbb{R}_+$ che soddisfi la relazione $f_n = O(n^2)$ ma non soddisfi la relazione $f_n = \Theta(n^2)$.

b) Esiste una sequenza $\{g_n\}_{n\in\mathbb{N}} \subseteq \mathbb{R}_+$ tale che $g_n = o(n^3)$ e $n^2 = o(g_n)$? In caso affermativo fornire un esempio; in caso contrario giustificare la risposta.

Esercizio 3.

Consideriamo l'algoritmo che su input $n \in \mathbb{N}$, $n > 2$, introduce in un albero di ricerca binaria inizialmente vuoto la seguente sequenza di numeri interi, utilizzando la tradizionale procedura di inserimento:

$$n, n-1, n+1, n-2, n+2, \ldots, 1, 2n-1, 0, 2n$$

a) Descrivere l'albero ottenuto assumendo il tradizionale ordinamento sugli interi.

b) Valutare in funzione di n l'ordine di grandezza del tempo di calcolo richiesto dall'algoritmo (assumendo il criterio uniforme).

c) Valutare l'ordine di grandezza del tempo di calcolo richiesto dall'algoritmo che su input $n \in \mathbb{N}$, $n > 2$, esegue lo stesso calcolo usando però gli alberi 2-3 invece degli alberi di ricerca binaria.

Esercizio 4.

Dato un albero binario $T =< V, E >$, sia $h : V \longrightarrow \mathbb{N}$ la funzione così definita:

$$h(v) = \begin{cases} 0 & \text{se } v \text{ è una foglia} \\ h(sin(v)) + 1 & \text{se } v \text{ possiede solo il figlio sinistro} \\ h(des(v)) & \text{se } v \text{ possiede solo il figlio destro} \\ \max\{h(sin(v)) + 1, h(des(v))\} & \text{se } v \text{ possiede sia il figlio sinistro che il figlio destro} \end{cases} \quad (1)$$

a) Definire una procedura che su input T (rappresentato mediante una coppia di vettori sin e des) calcoli i valori $h(v)$ per tutti i nodi v di T.

b) Valutare il tempo di calcolo e lo spazio di memoria richiesti dalla procedura descritta nel punto precedente.

ALGORITMI E STRUTTURE DATI
Prova scritta del 22.1.2007

Esercizio 1.

Eseguire l'algoritmo Quicksort sull'input $6, 7, 4, 2, 3, 1, 9, 8, 5, 0$ assumendo sempre come pivot il primo elemento della sequenza corrente e mettendo in evidenza gli scambi compiuti.

Esercizio 2.

Consideriamo l'algoritmo che su input $n \in \mathbb{N}$, $n > 2$, introduce in un albero di ricerca binaria inizialmente vuoto la seguente sequenza di numeri interi, utilizzando la tradizionale procedura di inserimento:

$$1, 2n, 2, 2n-1, 3, 2n-2, \dots, n, n+1$$

a) Descrivere l'albero ottenuto assumendo il tradizionale ordinamento sugli interi.

b) Valutare in funzione di n l'ordine di grandezza del tempo di calcolo richiesto dall'algoritmo.

Esercizio 3.

Considera la seguente procedura che esegue una sequenza di operazioni UNION-FIND su un insieme di $n > 2$ elementi $A = \{a_1, a_2, \dots, a_n\}$, a partire dalla partizione identità .

```
begin
      for i = 2, ..., n do
          if i pari   then UNION(a_i, a_1)
                      else UNION(a_1, a_i)
      t := FIND(a_1)
end
```

a) Supponendo di utilizzare una foresta semplice, determinare la foresta ottenuta al termine della procedura descritta. Valutare inoltre l'ordine di grandezza del tempo di calcolo richiesto in funzione del parametro n (si assuma il criterio uniforme).

b) Svolgere l'esercizio precedente supponendo di utilizzare una foresta con bilanciamento.

Esercizio 4.

Ricordiamo che in un grafo orientato G un nodo v è *raggiungibile* da un nodo s se esiste in G un cammino da s a v.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e due nodi $s, u \in V$.

Soluzione : la lista dei nodi che formano un cammino di lunghezza minima da s a u se u è raggiungibile da s, altrimenti il valore convenzionale $\perp$.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 9.1.2007

Esercizio 1.

Considera le sequenza $\{t_n\}_{n\in\mathbb{N}}$ così definita:

$$t_n = \begin{cases} e^n & \text{se } 0 \leq n \leq 30 \\ n^2\lceil\log n\rceil & \text{se } n > 30 \text{ e } n \text{ è pari} \\ 3n^2 & \text{se } n > 30 \text{ e } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ t_n = O(n^2\log n), \quad 2)\ t_n = \Theta(n^2\log n), \quad 3)\ t_n = o(n^2\log n), \quad 4)\ t_n \sim 3n^2$$

Esercizio 2.

a) Definire la nozione di sistema di indipendenza e quella di matroide.
b) Fornire un esempio di matroide.
c) Fornire un esempio di sistema di indipendenza che non è un matroide.

Esercizio 3.

Eseguire la seguente sequenza di operazioni *Union-Find* sull'insieme $\{2,3,4,5,6,7,8,9\}$ a partire dalla partizione iniziale (identità) usando una rappresentazione ad albero con bilanciamento: Union$(3,4)$, Union$(5,6)$, Union$(3,5)$, Union$(2,3)$, Union$(9,7)$, Union$(7,8)$, Find(3), Find(6), Union$(7,3)$, Find(9).

Per convenzione, nell'esecuzione delle operazioni Union, a parità di dimensione dei due alberi, la radice del nuovo insieme sia quella di minor valore. Si metta in evidenza la foresta che si ottiene dopo l'esecuzione di ogni operazione.

Esercizio 4.

Dato un albero ordinato $T = (V, E)$, per ogni nodo $v \in V$ denotiamo con $D(v)$ l'insieme dei nodi del sottoalbero di radice v. Inoltre, fissata una funzione peso $p : V \longrightarrow \mathbb{N}$, definiamo la nuova funzione $m_p : V \longrightarrow \mathbb{N}$ tale che, per ogni nodo $v \in V$,

$$m_p(v) = \begin{cases} p(v) & \text{se } v \text{ e' una foglia} \\ \min\{p(u) \mid u \in D(v)\} & \text{altrimenti} \end{cases}$$

a) Definire un algoritmo per il seguente problema:

Istanza : un albero ordinato T, rappresentato mediante liste di adiacenza, e per ogni nodo v di T il peso $p(v) \in \mathbb{N}$.
Soluzione : per ogni nodo v di T il valore $m_p(v)$.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti in funzione del numero n di nodi dell'albero (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 6.9.2006

Esercizio 1.

Determinare, al crescere di $n \in \mathbb{N}$ a $+\infty$, l'espressione asintotica di

$$\sum_{i=0}^{\lfloor \log n \rfloor} \log \frac{n^2}{2^i}$$

supponendo che il logaritmo sia in base 2.

Esercizio 2.

Dato un albero con radice T, con insieme dei nodi V, e una funzione peso $p : V \longrightarrow \mathbb{R}$, chiamiamo *costo* di un nodo $v \in V$ il valore

$$c(v) = \begin{cases} p(v) & \text{se } v \text{ e' la radice} \\ p(v) + c(padre(v)) & \text{altrimenti} \end{cases}$$

a) Definire un algoritmo per il seguente problema:

Istanza : un albero con radice T, rappresentato mediante liste di adiacenza, e per ogni nodo v di T il peso $p(v) \in \mathbb{R}$.
Soluzione : l'insieme delle foglie di T che hanno costo minimo.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti in funzione del numero n di nodi dell'albero (si assuma il criterio uniforme).

Esercizio 3.

a) Descrivere un algoritmo del tipo *divide et impera* che su input $n, k, a_1, a_2, \ldots, a_n \in \mathbb{N}$ calcoli il valore

$$a_1^k + a_2^k + \cdots + a_n^k \ .$$

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione dei parametri n e k.

ALGORITMI E STRUTTURE DATI
Prova scritta del 4.7.2006

Esercizio 1.

Sia x un numero reale maggiore di 1. Verificare se, al crescere di $n \in \mathbb{N}$, vale la relazione

$$\sum_{i=0}^{n} ix^i = \Theta(nx^n)$$

giustificando la risposta.

Esercizio 2.

a) Definire la nozione di albero e quella di albero ordinato.
b) Definire la nozione di albero binario e discutere la differenza tra alberi ordinati e alberi binari.

Esercizio 3.

a) Definire la nozione di foresta.
b) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo non orientato $G = \langle V, E\rangle$ rappresentato mediante liste di adiacenza.
Soluzione : il valore 1 se G è una foresta, il valore 0 altrimenti.

c) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

Esercizio 4.

a) Definire la struttura dati associata alle operazioni UNION, FIND.
b) Presentare una implementazione di tale struttura.

ALGORITMI E STRUTTURE DATI
Prova scritta del 1.6.2006

Esercizio 1.

a) Definire la nozione di sistema di indipendenza e quella di matroide.
b) Fornire un esempio di matroide.
c) Fornire un esempio di sistema di indipendenza che non è un matroide.

Esercizio 2.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza.
Soluzione : il valore 1 se G possiede un ciclo, il valore 0 altrimenti.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

Esercizio 3.

a) Descrivere un algoritmo del tipo *divide et impera* che su input $n, k, b_1, b_2, \dots, b_n \in \mathbb{N}$ calcoli il valore

$$F(n, k, \underline{b}) = k^{b_1} + k^{b_2} + \cdots + k^{b_n} \ .$$

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti nel caso peggiore in funzione dei parametri n e $h = \max\{b_i \mid i = 1, 2, \dots, n\}$.

ALGORITMI E STRUTTURE DATI
Prova scritta del 4.4.2006

Esercizio 1.

Eseguire l'algoritmo Heapsort sull'input

$$3, 1, 4, 5, 2, 7, 4, 8, 0, 6$$

mettendo in evidenza gli scambi eseguiti.

Esercizio 2.

a) Descrivere un algoritmo del tipo *divide et impera* che su input $n, k, b_1, b_2, \ldots, b_n \in \mathbb{N}$ calcola il valore

$$F(n, k, \underline{b}) = b_1^k + b_2^k + \cdots + b_n^k \ .$$

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione dei parametri n e k.

Esercizio 3.

Consideriamo un albero con radice $T =< V, E >$ nel quale i lati sono pesati mediante una funzione $w : E \longrightarrow \mathbb{R}_+$. Quindi, per ogni lato $\ell \in E$, $w(\ell)$ è un numero reale positivo che rappresenta il peso di ℓ. Inoltre, dato un cammino $C = (v_1, v_2, \ldots, v_k)$ in T, chiamiamo *peso* di C il valore

$$w(C) = \sum_{i=1}^{k-1} w(v_i, v_{i+1}) \ .$$

a) Definire un algoritmo che risolve il seguente problema mediante una visita in profondità :

Istanza : un albero con radice $T =< V, E >$ nel quale r denota la radice, rappresentato mediante liste di adiacenza, e una funzione peso $w : E \longrightarrow \mathbb{R}_+$ i cui valori sono inclusi nelle liste di adiacenza.

Soluzione : l'insieme delle coppie (u, p_u) dove u è una foglia dell'albero T e p_u è il peso del cammino che va da r a u.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti in funzione del numero n di nodi dell'albero (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 7.2.2006

Esercizio 1.

Determinare, in funzione di $n \to +\infty$, l'espressione asintotica della somma

$$\sum_{i=0}^{n-1} \frac{i}{n-i}$$

giustificando la risposta.

Esercizio 2.

Dato un intero positivo n, diamo le seguenti definizioni:

1. sia $U = \{1, 2, \ldots, n\}$;
2. sia $c : U \longrightarrow \mathbb{Z}$ una funzione costo, dove $\mathbb{Z}$ è l'insieme dei numeri interi relativi;
3. per ogni $A \subseteq U$, sia $c(A) = \sum_{i \in A} c(i)$;
4. dato un valore $H > 0$, sia $C_H = \{A \subseteq U \mid c(A) \leq H\}$.

Rispondere alle seguenti domande giustificando le risposte:

a) La coppia $\langle U, C_H \rangle$ forma un sistema di indipendenza?

b) Supponiamo ora che la funzione c assuma solo valori positivi, cioè $c : U \longrightarrow \mathbb{N}$. In questo caso $\langle U, C_H \rangle$ è un sistema di indipendenza? Nelle stesse ipotesi $\langle U, C_H \rangle$ è un matroide?

c) Consideriamo ora una nuova funzione valore $v : U \longrightarrow \mathbb{N}$ e supponiamo sempre che c assuma solo valori positivi. Definire una procedura greedy per determinare un sottoinsieme di valore massimo in C_H su input n, H, c, v.

d) L'algoritmo determina sempre un elemento di C_H di valore massimo?

e) Assumendo il criterio di costo uniforme, valutare in funzione di n l'ordine di grandezza del tempo di calcolo richiesto dall'algoritmo nel caso peggiore.

Esercizio 3.

Dati tre interi $a, b, n \in \mathbb{N}$ considera l'espressione

$$F(a,b,n) = \begin{cases} 1 & \text{se } n = 0 \\ a^n + a^{n-1}b + a^{n-2}b^2 + \cdots + ab^{n-1} + b^n & \text{se } n > 1 \end{cases}$$

a) Descrivere un algoritmo del tipo *divide et impera* che calcola $F(a, b, n)$ su input a, b, n.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n.

ALGORITMI E STRUTTURE DATI
Prova scritta del 16.12.2005
TEMA N. 1

Esercizio 1.
Eseguire l'algoritmo Heapsort sull'input

$$1, 3, 2, 6, 4, 0, 8, 7$$

mettendo in evidenza gli scambi compiuti.

Esercizio 2.
Determinare l'espressione asintotica di $\log((2n)!)$.

Esercizio 3.
Considera il problema di calcolare l'espressione

$$(a_1 + 3a_2 + a_3 + \cdots + (2 + (-1)^n)a_n) \pmod k$$

assumendo come input la sequenza di numeri interi $k, n, a_1, a_2, \ldots, a_n$ (dove $k, n \geq 1$).
a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.
b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n.

Esercizio 4.
a) Descrivere una procedura ricorsiva per risolvere il seguente problema:

Istanza : un albero con radice T di n nodi, rappresentato mediante liste di adiacenza.
Soluzione : il numero di discendenti di ciascun nodo di T.

b) Descrivere una procedura non ricorsiva per risolvere lo stesso problema.

ALGORITMI E STRUTTURE DATI
Prova scritta del 16.12.2005
TEMA N. 2

Esercizio 1.
Determinare l'espressione asintotica di $\log((n^2)!)$.

Esercizio 2.
Eseguire l'algoritmo Quicksort sull'input

$$5, 6, 4, 1, 3, 0, 8, 7, 2, 9$$

assumendo sempre come pivot il primo elemento della sequenza corrente e mettendo in evidenza gli scambi compiuti.

Esercizio 3.
a) Descrivere una procedura per risolvere il seguente problema:

Istanza : un grafo orientato $G = (V, E)$ di n nodi e m lati, rappresentato mediante liste di adiacenza e due nodi $s, u \in V$.
Soluzione : l'insieme dei nodi G che si trovano alla stessa distanza da s e da u.

b) Assumendo il criterio di costo uniforme, valutare in funzione di n e m il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo nel caso peggiore.

Esercizio 4.
Considera il problema di calcolare l'espressione

$$(3a_1 + 5a_2 + 3a_3 + \cdots + (4 + (-1)^n)a_n) \pmod{k}$$

assumendo come input la sequenza di numeri interi $k, n, a_1, a_2, \ldots, a_n$ (dove $k, n \geq 1$).
a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.
b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n.

ALGORITMI E STRUTTURE DATI

Svolgimento della prova scritta del 15.9.2005

Esercizio 1.

Eseguire l'algoritmo Quicksort sull'input

$$5, 2, 6, 9, 4, 6, 8, 7, 1, 4$$

mettendo in evidenza gli scambi eseguiti e assumendo sempre come pivot il primo elemento della sequenza corrente.

Esercizio 2.

Dato un intero positivo k, per ogni sequenza finita $a_1, a_2, \ldots, a_n$ di interi relativi definiamo il valore

$$F(a_1, a_2, \ldots, a_n) = (3a_1 + 5a_2 + 3a_3 + \cdots + (4 + (-1)^n)a_n) \mod k$$

a) Definire una procedura del tipo *Divide et Impera* per calcolare $F(a_1, a_2, \ldots, a_n)$ su input $k, n, a_1, a_2, \ldots, a_n$.

b) Assumendo il criterio di costo uniforme valutare il tempo di calcolo e lo spazio di memoria richiesti dalla procedura descritta nel punto precendente.

Esercizio 3. Considera le sequenza $\{t_n\}$ così definita:

$$t_n = \begin{cases} 2n + 4n^3\lceil \log n \rceil & \text{se } n \text{ è pari} \\ n^2\lfloor \log n \rfloor + 2\lfloor e^n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ t_n = O(n^3 \log n), \quad 2)\ t_n = O(e^n), \quad 3)\ e^n = O(t_n), \quad 4)\ t_n = \Theta(e^n), \quad 5)\ n^3 = o(t_n).$$

Esercizio 4.

a) Descrivere un algoritmo ricorsivo per risolvere il seguente problema:

Istanza : un albero ordinato T rappresentato mediante liste di adiacenza;
Soluzione : il numero di discendenti di ogni nodo di T

b) Descrivere un algoritmo non ricorsivo per risolvere lo stesso problema.

ALGORITMI E STRUTTURE DATI
Prova scritta del 1^o.7.2005
TEMA N. 1

Esercizio 1.

Considera il problema di calcolare l'espressione

$$(k^n)mod(13)$$

assumendo come input due interi positivi n, k tali che $13 \leq k$.

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria in funzione di n.

c) Assumiamo il criterio di costo logaritmico. Determinare una stima O-grande del tempo di calcolo e dello spazio di memoria in funzione dei parametri n e k.

Esercizio 2.

a) Spiegare la differenza tra ordine di grandezza di una sequenza $\{f_n\} \subseteq \mathbb{R}_+$ ed espressione asintotica di $\{f_n\}$.

b) Determinare l'ordine di grandezza della sequenza $\{g_n\}$ definita da

$$g_n = \sum_{i=1}^{n} (5i^{2/3} + 1) \log i$$

c) Determinare una espressione asintotica di $\{g_n\}$.

Esercizio 3.

Ricordiamo che in un grafo orientato G un nodo v è *raggiungibile* da un nodo s se esiste in G un cammino da s a v.

a) Descrivere un algoritmo per il seguente problema:

> Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e due nodi $s, u \in V$.
>
> Soluzione : il valore convenzionale $\perp$ se u non è raggiungibile da s, altrimenti la lista dei lati T che formano un cammino di lunghezza minima da s a u.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Prova scritta del 1^o.6.2005
TEMA N. 1

Esercizio 1.
Eseguire l'algoritmo Quicksort sull'input

$$4, 2, 6, 0, 1, 7, 8, 9, 3, 5$$

mettendo in evidenza gli scambi eseguiti e assumendo sempre come pivot il primo elemento della sequenza corrente.

Esercizio 2.
a) Descrivere una procedura per risolvere il seguente problema:

Istanza : un grafo orientato G di n nodi e m lati, rappresentato mediante liste di adiacenza;
Soluzione : la lunghezza del più lungo cammino semplice in G.

b) Assumendo il criterio di costo uniforme, valutare in funzione di n e m il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo nel caso peggiore.

Esercizio 3.
Considera le sequenze $\{f_n\}$ e $\{g_n\}$ così definite:

$$f_n = \sum_{i=1}^{n} (2i^2 + 1) \log i \qquad\qquad g_n = \sum_{i=1}^{n} \frac{5}{i}$$

Determinare l'ordine di grandezza di $\{f_n\}$ e $\{g_n\}$.

ALGORITMI E STRUTTURE DATI
Prova scritta del 5.4.2005

Esercizio 1.
Considera la sequenze $\{f_n\}$ e così definita:

$$f_n \quad = \quad \begin{cases} 5n^3\lfloor \log n \rfloor + n & \text{se } n \text{ e' divisibile per 3} \\ 3(2^n/n^2) + 5n^3\lfloor \log n \rfloor & \text{altrimenti} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1) f_n \sim 5n^3 \log n, \quad 2) f_n = \mathrm{O}(2^n), \quad 3) f_n = \Theta(2^n/n^2), \quad 4) f_n = \mathrm{O}(n^4), \quad 5) f_n = o(2^n/n).$$

Esercizio 2.
Dato un albero binario $T =< V, E >$, sia $h_T : V \longrightarrow \mathbb{Z}$ la funzione così definita:

$$h_T(v) = \begin{cases} 1 & \text{se } v \text{ è una foglia} \\ 2h_T(sin(v)) & \text{se } v \text{ possiede solo il figlio sinistro} \\ 2h_T(des(v)) & \text{se } v \text{ possiede solo il figlio destro} \\ h_T(sin(v)) + h_T(des(v)) & \text{se } v \text{ possiede sia il figlio sinistro che il figlio destro} \end{cases} \tag{2}$$

a) Definire una procedura ricorsiva che su input T (rappresentato mediante una coppia di vettori *sin* e *des*) calcoli i valori $h_T(v)$ per tutti i nodi v di T.

b) Valutare il tempo di calcolo e lo spazio di memoria richiesti dalla procedura descritta nel punto precedente.

Esercizio 3.
Dati due interi positivi k, n e una sequenza $a_1, a_2, \ldots, a_n$ tale che $a_i \in \mathbb{Z}$ per ogni $i = 1, 2, \ldots, n$, conssidera il valore

$$G(a_1, a_2, \ldots, a_n) = \left(2a_1 + a_2 + 2a_3 + \cdots + \frac{(3 - (-1)^n)}{2} a_n\right) \bmod k$$

a) Definire una procedura del tipo *Divide et Impera* per calcolare $G(a_1, a_2, \ldots, a_n)$ su input $k, n, a_1, a_2, \ldots, a_n$.

b) Assumendo il criterio di costo uniforme valutare il tempo di calcolo e lo spazio di memoria richiesti dalla procedura descritta nel punto precendente.

c) Supponendo che ogni a_i sia un intero di m bit, valutare il tempo di calcolo richiesto dalla procedura assumendo il criterio di costo logaritmico.

ALGORITMI E STRUTTURE DATI
Prova scritta del 8.2.2005

Esercizio 1.

Eseguire l'algoritmo Heapsort sull'input

$$3, 1, 5, 6, 2, 8, 4, 9, 0, 7$$

mettendo in evidenza gli scambi eseguiti.

Esercizio 2.

Dato un grafo orientato, diciamo che un nodo v è *raggiungibile* da un nodo s se esiste un cammino da s a v e, in questo caso, chiamiamo *distanza* di v da s la lunghezza del più corto cammino che va da s a v.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza, un nodo $r \in V$ e un intero positivo k.

Soluzione : l'insieme dei nodi in V raggiungibili da r che si trovano a distanza maggiore o uguale a k da r.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

Esercizio 2.

Considera il problema di calcolare l'espressione

$$(a^n) mod(k)$$

assumendo come input tre interi positivi a, n e k.

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria in funzione di n.

c) Assumiamo il criterio di costo logaritmico. Determinare una stima O-grande del tempo di calcolo e dello spazio di memoria in funzione dei parametri a, n e k.

ALGORITMI E STRUTTURE DATI
Prova scritta del 14.1.2005

Esercizio 1.

Eseguire l'algoritmo Quicksort sull'input

$$4, 2, 6, 0, 1, 7, 8, 9, 3, 5$$

mettendo in evidenza gli scambi eseguiti e assumendo sempre come pivot il primo elemento della sequenza corrente.

Esercizio 2.

a) Definire la nozione di albero di ricerca binaria e descrivere ad alto livello due procedure ricorsive, la prima per la ricerca di un elemento in tale struttura e l'altra per l'inserimento di un nuovo elemento.

b) Descrivere ad alto livello due procedure *non ricorsive* per eseguire le medesime operazioni.

Esercizio 3.

a) Descrivere una procedura del tipo *divide et impera* per il calcolo dell'espressione

$$X = a_1 \cdot a_2 + a_2 \cdot a_3 + a_3 \cdot a_4 + \cdots + a_{n-1} \cdot a_n$$

su input $n, a_1, a_2, \ldots, a_n$.

b) Assumendo il criterio di costo uniforme, valutare in funzione di n il tempo di calcolo e lo spazio di memoria richiesti dalla procedura su input $n, a_1, a_2, \ldots, a_n$.

c) Supponendo che ogni a_i sia un intero di k bit, fornire una valutazione O-grande del tempo di calcolo e dello spazio di memoria richiesti dalla procedura secondo il criterio logaritmico, in funzione dei parametri n e k.

ALGORITMI E STRUTTURE DATI
Prova scritta del 12.11.2004
TEMA N. 1

Esercizio 1.
Considera le sequenze $\{f_n\}$ e $\{g_n\}$ così definite:

$$f_n = \sum_{i=1}^{n} (i^2 + 3) \log i \qquad\qquad g_n = \sum_{i=1}^{n} \frac{2}{i}$$

Determinare l'ordine di grandezza di $\{f_n\}$ e $\{g_n\}$.

Esercizio 2.
Considera il problema di calcolare l'espressione

$$na_1 + (n-1)a_2 + (n-2)a_3 + \cdots + 2a_{n-1} + a_n$$

assumendo come input la sequenza di numeri interi $n, a_1, a_2, \ldots, a_n$ (dove $n \geq 1$).
a) Descrivere un algoritmo per risolvere il problema.
b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n.
c) Assumiamo il criterio di costo logaritmico e supponiamo che ogni a_i sia un intero di k bits. Valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dall'algoritmo in funzione dei parametri n e k.

Esercizio 3.
a) Descrivere una procedura per risolvere il seguente problema:

Istanza : un grafo orientato G di n nodi e m lati, rappresentato mediante liste di adiacenza, un nodo s di G e un intero positivo k;
Soluzione : il numero di nodi di G che si trovano a distanza k da s.

b) Assumendo il criterio di costo uniforme, valutare in funzione di n e m il tempo di calcolo e lo spazio di memoria richiesti dall'algoritmo nel caso peggiore.

ALGORITMI E STRUTTURE DATI
Prova scritta del 12.11.2004
TEMA N. 2

Esercizio 1.
Considera la sequenza $\{f_n\}$ così definita:

$$f_n \quad = \quad \begin{cases} n \log n + 2n^2 & \text{se } n \equiv 0 \pmod 3 \\ 3n^3 \log\left(1 + \frac{2}{n}\right) & \text{se } n \equiv 1 \pmod 3 \\ 2n \log n + \frac{n^2}{3}\left(1 + \frac{2}{n}\right)^n & \text{se } n \equiv 2 \pmod 3 \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1) f_n \sim 2n^2, \quad 2) f_n = O(n^3), \quad 3) f_n = \Theta(n^2), \quad 4) f_n = o(n^2), \quad 5) f_n = O(n \log n).$$

Esercizio 2.
Considera il seguente problema:

Istanza : un intero $n \geq 1$ e una sequenza di $2n$ numeri interi $a_1, a_2, \ldots, a_{2n}$.
Soluzione : il numero delle coppie (a_i, a_{i+n}) tali che $a_i = a_{i+n}$ e $i \in \{1, 2, \ldots, n\}$.

a) Descrivere un algoritmo, il più efficiente possibile, per risolvere il problema.
b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti in funzione del parametro n.
c) Assumiamo il criterio di costo logaritmico e supponiamo che ogni a_i sia un intero di k bits. Valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dall'algoritmo in funzione dei parametri n e k.

Esercizio 3. Dato un albero ordinato T e un nodo v in T, chiamiamo *altezza minima* di v la minima distanza di v da una foglia. Chiamiamo *peso* di T la somma delle altezze minime dei suoi nodi.
a) Descrivere una procedura per risolvere il seguente problema:

Istanza : un albero ordinato T di n nodi, rappresentato mediante liste di adiacenza;
Soluzione : il peso di T.

b) Assumendo il criterio uniforme, valutare il tempo di calcolo richiesto in funzione del numero di nodi dell'albero.
c) A parità di nodi quali alberi hanno peso massimo e quali peso minimo?

ALGORITMI E STRUTTURE DATI
Tema d'esame del 8.1.2004

Esercizio 1.
Considera le sequenza $\{t_n\}$ così definita:

$$t_n = \begin{cases} 2\lfloor e^n \rfloor - n^2 \lfloor \log n \rfloor & \text{se } n \text{ è pari} \\ 3n^2 \lceil \log n \rceil - 4n & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ t_n = O(n^2 \log n), \quad 2)\ t_n = O(e^n), \quad 3)\ t_n = \Theta(e^n), \quad 4)\ t_n = o(n^3).$$

Esercizio 2.
Considera il seguente problema:

Istanza : due sequenze finite di numeri interi $a_0, a_1, \dots, a_n$, $b_0, b_1, \dots, b_n$.
Soluzione : il valore dell'espressione $c = (a_0 + b_n) \cdot (a_1 + b_{n-1}) \cdot \ \cdots \ \cdot (a_n + b_0)$.

a) Stimare il tempo di calcolo e lo spazio di memoria richiesti dalla seguente procedura assumendo il criterio di costo uniforme:

begin
  leggi e memorizza l'input
  $s = 1$
  for $i = 0, 1, \dots, n$ **do**
    $\begin{cases} t := a_{n-i} + b_i \\ s := s \cdot t \end{cases}$
  return s
end

b) Supponendo che ogni a_i e ogni b_i abbia dimensione m, svolgere l'esercizio precedente assumento il criterio di costo logaritmico.
c) Descrivere un algoritmo del tipo *divide et impera* per risolvere lo stesso problema.
d) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria richiesti dalla procedura descritta nell'esercizio precedente.

Esercizio 3.
Dato un grafo orientato, diciamo che un nodo v è *raggiungibile* da un nodo s se esiste un cammino da s a v e, in questo caso, chiamiamo *distanza* di v da s la lunghezza del più corto cammino che va da s a v.
a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e un nodo $s \in V$.
Soluzione : il numero dei nodi in V raggiungibili da s che si trovano alla massima distanza da s.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Tema d'esame del 5.9.2003

Esercizio 1.

Considera le sequenza $\{t_n\}$ così definita:

$$t_n = \begin{cases} n^2\lfloor \log n \rfloor + 2\lfloor e^n \rfloor & \text{se } n \text{ è pari} \\ 4n + 3n^2\lceil \log n \rceil & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ t_n = O(n^2 \log n), \quad 2)\ t_n = O(e^n), \quad 3)\ t_n = \Theta(e^n), \quad 4)\ t_n = o(n^3).$$

Esercizio 2.

a) Descrivere un algoritmo del tipo *divide et impera* per determinare la classe di resto modulo t del prodotto di una sequenza di numeri interi $a_1, a_2, \ldots, a_n$, ovvero l'espressione

$$[a_1 \cdot a_2 \cdot \ldots \cdot a_n](\mathrm{mod}\ t).$$

Si supponga che l'input dell'algoritmo sia dato dagli interi n e t seguiti dalla sequenza $a_1, a_2, \ldots, a_n$.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria in funzione di n.

c) Assumiamo il criterio di costo logaritmico e supponiamo che ogni a_i sia un intero di k bit. Determinare una stima O-grande del tempo di calcolo in funzione dei parametri n, t e k.

Esercizio 3.

Dato un grafo orientato, diciamo che un nodo v è *raggiungibile* da un nodo s se esiste un cammino da s a v e, in questo caso, chiamiamo *distanza* di v da s la lunghezza del più corto cammino che va da s a v.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza, un nodo $s \in V$ e un intero positivo k.

Soluzione : il numero dei nodi in V raggiungibili da s che si trovano a distanza k da s.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

ALGORITMI E STRUTTURE DATI
Tema d'esame del 1.7.2003

Esercizio 1.

Considera le sequenza $\{g_n\}$ così definita:

$$g_n = \begin{cases} 2n^2\lceil \log n \rceil + 5n & \text{se } n \text{ è pari} \\ 3\lfloor e^n \rfloor + 3n^2 \lfloor \log n \rfloor & \text{se } n \text{ è dispari} \end{cases}$$

Tra le seguenti relazioni determinare le relazioni vere e quelle false:

$$1)\ g_n = O(n^2 \log n), \quad 2)\ g_n = O(e^n), \quad 3)\ g_n = \Theta(e^n), \quad 4)\ g_n = o(n^3).$$

Esercizio 2.

Dato un grafo orientato, diciamo che un nodo v è *raggiungibile* da un nodo s se esiste un cammino da s a v e, in questo caso, chiamiamo *distanza* di v da s la lunghezza del più corto cammino che va da s a v.

a) Descrivere un algoritmo per il seguente problema:

Istanza : un grafo orientato $G = \langle V, E \rangle$ rappresentato mediante liste di adiacenza e un nodo $s \in V$.

Soluzione : il numero dei nodi in V raggiungibili da s che si trovano alla massima distanza da s.

b) Svolgere l'analisi dell'algoritmo valutando il tempo di calcolo e lo spazio di memoria richiesti su un input di n nodi e m lati (si assuma il criterio uniforme).

Esercizio 3.

a) Descrivere un algoritmo del tipo *divide et impera* per determinare l'elemento minimo e l'elemento massimo di una sequenza di numeri interi $a_1, a_2, \ldots, a_n$. Si supponga che l'input dell'algoritmo sia dato dall'intero n seguito dalla sequenza $a_1, a_2, \ldots, a_n$.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria in funzione di n.

c) Assumiamo il criterio di costo logaritmico e supponiamo che ogni a_i sia un intero di k bit. Determinare una stima O-grande del tempo di calcolo e dello spazio di memoria in funzione dei parametri n e k.

ALGORITMI E STRUTTURE DATI
Tema d'esame del 4.6.2003

Esercizio 1.

Eseguire l'algoritmo Mergesort sull'input

$$3, 4, 1, 5, 6, 0, 5, 3, 8, 9$$

mettendo in evidenza i confronti eseguiti.

Esercizio 2.

Considera il problema di calcolare l'espressione

$$(7^n) mod(k)$$

assumendo come input due interi positivi n, k, con $7 < k$.

a) Descrivere un algoritmo del tipo *divide et impera* per risolvere il problema.

b) Assumendo il criterio di costo uniforme, valutare l'ordine di grandezza del tempo di calcolo e dello spazio di memoria in funzione di n.

c) Assumiamo il criterio di costo logaritmico. Determinare una stima O-grande del tempo di calcolo e dello spazio di memoria in funzione dei parametri n e k.

Esercizio 3.

Dato un albero con radice T, per ogni nodo v in T denotiamo con $D(v)$ l'insieme dei discendenti di v; inoltre, per ogni $w \in D(v)$ denotiamo con $\ell(w)$ la lunghezza del cammino da v a w. Infine, chiamiamo *peso* del nodo v la somma $\sum_{w \in D(v)} \ell(w)$.

a) Descrivere una procedura ricorsiva che, avendo in input un albero con radice T (rappresentato mediante liste di adiacenza), calcoli il peso di ciascun nodo di T.

b) Assumendo il criterio di costo uniforme, valutare il tempo di calcolo e lo spazio di memoria richiesti in funzione del numero n di nodi dell'albero.

c) (facoltativo) Descrivere una procedura iterativa per risolvere lo stesso problema.