Tema d'esame di Algoritmi e Strutture Dati
Modulo A
19/06/2006

Tempo a disposizione: 3 ore.

1. **[5 punti]** Si dimostri la verità o la falsità della seguente affermazione:

$$\text{se } 2^{f(n)} = \Theta(2^{g(n)}), \text{ allora } f(n) = \Theta(g(n))$$

2. **[5 punti]** Sia data la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 3\,T(n/4) + \sqrt{n} & \text{se } n > 1 \end{cases}$$

   Risolvere l'equazione di ricorrenza utilizzando il metodo iterativo (alberi di ricorrenza).

3. **[10 punti]** Si scriva un *algoritmo ricorsivo* efficiente che cancelli da un albero binario di ricerca T tutte le chiavi contenute in un altro albero binario di ricerca T'.

4. **[10 punti]** Si scriva un algoritmo dettagliato che, dato un grafo orientato G, ne calcoli le *compinenti fortemente connesse* e stampi i vertici contenuti in ciascuna componente e il numero delle componenti del grafo.

Tema d'esame di Algoritmi e Strutture Dati I
11/09/2006

Tempo a disposizione: 3 ore.

1. **(4 punti)** Dimostrare o falsificare la seguente affermazione:

$$\textit{Se} \quad f(n) = O(g(n)) \quad \textit{allora} \quad \sqrt{g(n)} = \Omega(\sqrt{f(n)})$$

2. **(6 punti)** Risolvere la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n \leq 1 \\ 4T(n/9) + \sqrt{n} & \text{se } n > 1 \end{cases}$$

3. **(10 punti)** Si definisca un algoritmo ricorsivo efficiente che, ricevuti in ingresso un (riferimento ad un) Albero Binario di Ricerca T e tre valori k_{min}, k_{max} (con $k_{min} \leq k_{max}$) e $z \geq 0$ cancelli dall'albero T tutti i nodi che o hanno chiave con valore **esterno all'intervallo** $[k_{min}, k_{max}]$ o stanno a **distanza maggiore uguale a** z dalla radice. Non è ammesso l'uso di variabili globali né del passaggio di parametri per riferimento.

aciclico

4. **(10 punti)** Scrivere un algoritmo che, in tempo lineare sulla dimensione del grafo in input, ne calcoli un ordinamento topologico. **Non è ammesso l'impiego della visista in profondità (DFS).**

Tema d'esame di Algoritmi e Strutture Dati I
25/06/2007

Tempo a disposizione: 3 ore.

1. [6 punti] Si supponga che $h(n) = \Theta(t(n))$ e che $f(n)/h(n) = \Theta(g(n))$. Si dimostri la verità o la falsità della seguente affermazione:

$$\frac{f^2(n)}{h(n)\, t(n)} = \Theta(g^2(n))$$

2. [7 punti] Sia data la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 15\,T(n/4) + n^2 \log n & \text{se } n > 1 \end{cases}$$

Trovare la stima asintotica più vicina possibile a $T(n)$.

3. [11 punti] Scrivere un algoritmo ricorsivo efficiente che, dato un albero binario di ricerca T e un due valori k_1 e k_2 (con $k_1 \leq k_2$), cancelli da T tutti nodi con chiavi comprese tra k_1 e k_2 e restituisca il numero di nodi cancellati dall'albero.

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. [6 punti] Un *tour di Eulero* in un grafo orientato $G = \langle V, E \rangle$ è un percorso ciclico che attraversa ogni arco una e una sola volta. Un tour di Eulero in un grafo orientato G esiste se il grado entrante di ogni vertice è uguale al suo grado uscente. Si scriva un algoritmo che, dato un grafo orientato G, verifichi se esista o meno un tour di Eulero in tempo $\Theta(|V| + |E|)$.

Tema d'esame di Algoritmi e Strutture Dati I
16/07/2007

Tempo a disposizione: 3 ore.

1. [7 punti] Usando la definizione di Θ, si domostri la verità o la falsità della seguente affermazione:

$$\text{se } f(n) = \Theta(n) \text{ e } g(n) = \Theta(2^{n^2})), \text{ allora } 2^{2 \cdot log f(n)} = \Theta(log(g(n)))$$

2. [7 punti] Sia data la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 2 \cdot T(n/2) + T(n/4) + n & \text{se } n > 1 \end{cases}$$

Trovare la stima asintotica più vicina possibile a $T(n)$.

3. [10 punti] Scrivere un algoritmo ricorsivo efficiente che, dato un albero binario di ricerca T, due valori k_1 e k_2 (con $k_1 \leq k_2$) e un valore c, restituisca (se esiste), effettuando una sola visita dell'albero, il puntatore al nodo di T che ha chave compresa tra k_1 e k_2 e al tempo stesso che sia la più vicina possibile al (ma diversa dal) valore c.

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. [6 punti] Si scriva un algoritmo che, dato un grafo orientato G e un vertice v di G, verifichi in tempo lineare sulla dimensione del grafo se G è un albero radicato in v.

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

Tema d'esame di Algoritmi e Strutture Dati I (12/02/2008)

Tempo a disposizione: 3 ore.

1. **[7 punti]** Siano $f(n)$ e $g(n)$ due funzioni asintoticamente positive e crescenti. Dimostrare la verità o la falsità della seguente affermazione:

$$log\,(f(n) \cdot g(n)) = O(max\{log\,f(n), log\,g(n)\})$$

2. **[6 punti]** Si consideri la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 17\,T(n/2) + n^4 & \text{se } n > 1 \end{cases}$$

   Trovare la stima più vicina possibile a $T(n)$.

3. **[10 punti]** Si consideri un albero binario T che soddisfa la seguente proprietà: *il valore della chiave di ogni nodo è* **non minore** *del valore delle chiavi dei suoi due figli*.

   Si definisca un algoritmo ricorsivo che dato (il riferimento a) l'albero T e due valori di chiave k_{min} e k_{max} (con $k_{min} < k_{max}$), cancelli dall'albero T tutti i nodi con chiave compresa tra k_{min} e k_{max}, preservando la proprietà sopra riportata.

   Suggerimento: Può risultare utile sviluppare un alrgoritmo ricorsivo di appoggio che esegua la cancellazione della radice di un (sotto) albero del tipo descritto sopra e ritorni l'albero risultante.

4. **[7 punti]** Si definisca un algoritmo che, dato in ingresso un grafo G arbitrario e un vertice qualsiasi v, trasformi il grafo in ingresso nell'albero radicato in v dei percorsi minimi che si dipartono da v.

Tema d'esame di Algoritmi e Strutture Dati I
27/03/2008

Tempo a disposizione: 3 ore.

1. **(7 punti)** Saino f e g due funzioni positive e asintoticamente crescenti. A seconda del caso, dimostrare la verità o la falsità della seguenti affermazioni:

   (a) se $h^2(n) = \Theta(min\{f(n), g(n)\})$, allora $\sqrt{f(n)} = O(h(n))$.

   (b) se $\sqrt{h(n)} = O(min\{f(n), g(n)\})$, allora $h(n) = O(g^2(n))$.

2. **(6 punti)** Risolvere la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n \leq 1 \\ T(3n/4) + T(n/2) + n & \text{altrimenti} \end{cases}$$

3. **(9 punti)** Si definisca un algoritmo ricorsivo efficiente che, ricevuti in ingresso un (riferimento ad un) Albero Binario di Ricerca T e tre valori k_{min}, k_{max} (con $k_{min} \leq k_{max}$) e k, restituisca, se esiste, quel nodo di T che ha chiave con valore **interno all'intervallo** $[k_{min}, k_{max}]$ e che, contemporaneamente, sia **il più lontano possibile da** k. Non è ammesso l'uso di variabili globali né del passaggio di parametri per riferimento.

4. **(8 punti)** Scrivere un algoritmo che dato in ingresso un vertice s e un grafo orientato G, rappresentato con liste di adiacenza, stampi tutti percorsi ciclici di G che si dipartono da s.

Tema d'esame di Algoritmi e Strutture Dati I
20/06/2008

Tempo a disposizione: 3 ore.

1. [6 punti] Si domostri la verità o la falsità della seguente affermazione:

   se $2^{f(n)} = \Theta(g(n))$ e $g(n) = \Theta(h(n)^k)$ per una qualche costante $k > 0$, allora $f(n) = \Theta(\log h(n))$

2. [5 punti] Sia data la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n = 1 \\ 4T(n/4) + \sqrt{n} & \text{se } n \geq 2 \end{cases}$$

   Trovare la stima asintotica più vicina possibile a $T(n)$.

3. [10 punti] Sia dato un albero binario di ricerca T in cui ogni nodo contiene esclusivamente un campo per la chiave, uno per il puntatore al figlio destro e uno per il figlio sinistro. Siano inoltre dati in ingresso un possibile valore di chiave k e due valori l_1 e l_2 (con $l_1 \leq l_2$). Scrivere un algoritmo ricorsivo efficiente che cerchi e stacchi, se esiste, quel "**nodo dell'albero T che, tra i nodi che si trovano ad un livello di profondità esterno all'intervallo $[l_1, l_2]$ e che sono diversi dalla radice di T, contiene la chiave più vicina possibile a k ma maggiore di k**".

   L'algoritmo dovrà ritornare il riferimento al nodo cercato, se esso esiste. Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. [9 punti] Sia dato un grafo orientato $G = \langle V, E \rangle$, rappresentato tramite liste di adiacenza. Si definisca un algoritmo che verifichi in tempo lineare sulla dimensione del grafo se è vero che esiste un vertice $v \in V$ tale che:

   - se $u \in V$ allora $u \stackrel{\pi}{\leadsto} v$ per qualche percorso π;
   - per ogni $z \in V$, $v \stackrel{\lambda}{\leadsto} z$ per qualche percorso λ;

Tema d'esame di Algoritmi e Strutture Dati I
(12/01/2009)

Tempo a disposizione: 3 ore.

1. [6 punti] Assumento che f e g siano funzioni asintoticamente positive e crescenti, si dimostri la verità o la falsità della seguente affermazione:

$$\text{se } f(n) = \Theta(n) \text{ e } g(n) = \Theta(2^n), \text{ allora } 2^{f(n)} = \Theta(g(n))$$

2. [6 punti] Risolvere la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n \leq 1 \\ 6\,T(n/8) + \sqrt{n} & \text{se } n > 1 \end{cases}$$

3. [10 punti] Scrivere un algoritmo ricorsivo efficiente che, dato un albero binario di ricerca T, un possibile valore di chiave k, e due interi l_1 e l_2 (con $1 \leq l_1 \leq l_2$), cerchi un nodo dell'albero che soddisfa la seguente proprietà:

> "contiene la più grande chiave pari minore di k tra i nodi che si trovano a profondità compresa tra l_1 e l_2"

Se il nodo cercato esiste, l'algoritmo dovrà staccare dall'albero T il nodo trovato e ritornare il riferimento al nodo stesso. Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. [8 punti] Si definisca un algoritmo che, dato in ingresso un grafo G arbitrario e un vertice qualsiasi v, trasformi il grafo in ingresso nell'albero radicato in v dei percorsi minimi che si dipartono da v. L'algoritmo deve risolvere il problema in tempo lineare sulla dimensione del grafo.

Tema d'esame di Algoritmi e Strutture Dati I
15/07/2009

Tempo a disposizione: 3 ore.

1. Siano f e g due funzioni asintoticamente positive e crescenti. A seconda del caso, dimostrare la verità o la falsità delle seguenti affermazioni:

   (a) se $\sqrt{h(n)} = O(2^{(f(n)^{g(n)})})$, allora $\log\log h(n) = \Theta(g(n) \cdot \log f(n))$.

   (b) se $h(n) = \Theta(\max\{\log\log f(n), \log\log g(n)\})$, allora $g(n) = O\left(2^{(2^{h(n)})}\right)$.

2. Risolvere la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n \leq 2 \\ T(n/3) + T(n/4) + n & \text{altrimenti} \end{cases}$$

3. Si definisca un algoritmo ricorsivo efficiente che, noti un Albero Binario di Ricerca T, i valori k_{min}, k_{max} (con $k_{min} \leq k_{max}$) e un intero $x \geq 0$, cancelli tutti i nodi con chiave pari compresa tra k_{min} e k_{max} che sono radici di sottoalberi di altezza non infeririore ad x. Notare che il vincolo sulle altezze dei sottoalberi va misurato rispetto all'albero originario, non **a quello eventualmente modificato**. Non è ammesso l'uso di variabili globali, né del passaggio di parametri per riferimento.

4. Scrivere un algoritmo che dato in ingresso un vertice s e un grafo orientato G, rappresentato con liste di adiacenza, stampi tutti percorsi di G che raggiungono il vertice s.

OK

Tema d'esame di Algoritmi e Strutture Dati I
(Gruppo I)
22/01/2010

Tempo a disposizione: 3 ore.

1. Siano $f(n)$ e $g(n)$ due funzioni asintoticamente positive e crescenti. Dimostrare la verità o la falsità della seguente affermazione:

$$log\,(f(n) \cdot g(n)) = O(max\{log\,f(n), log\,g(n)\})$$

2. Risolvere la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} 1 & \text{se } n \leq 1 \\ 7T(n/2) + n^3 \log n & \text{se } n > 1 \end{cases}$$

3. Scrivere un algoritmo ricorsivo efficiente che cancelli da un Albero Binario di Ricerca T (i cui nodi contengono solo il campo chiave, figlio destro e figlio sinistro) ogni nodo, diverso dalla radice dell'albero, che soddisfa la sequente proprietà:

   "contiene una chiave pari minore di k ed è radice di un sottoalbero di altezza minore di H"

   dove H è un valore fornito in ingresso. Si noti che la proprietà dei nodi da cancellare è da intendersi rispetto all'albero originario T in ingresso. Non è ammesso l'uso di variabili globali né di passaggio di parametri per riferimento.

4. Dato un grafo orientato G e un vertice s di G, scrivere un algoritmo che in tempo lineare sulla dimensione del grafo, stampi, senza ripetizioni, tutti i vertici di G che o raggiungono s o sono da s raggiungibili.

Tema d'esame di Algoritmi e Strutture Dati I
22/06/2011

Tempo a disposizione: 3 ore.

1. Dimostrare per esteso, ricavando le costanti necessarie, le relazioni asintotiche sotto riportate:

   (a) $n\,log_2\,n - 3\,n - 18 = \Omega(n)$;

   (b) $3\,n^2 + 2\,n + 3 = \Theta(n^2 - 7)$.

2. Si derivi, mostrando per esteso il procedimento seguito, l'equazione di ricorrenza per la funzione $T(n)$ che descrive il tempo di esecuzione dell'algoritmo sotto riportato. Si risolva, poi, l'equazione, individuando la stima asintotica più vicina possibile a $T(n)$.

   1 ALGO(n)

   ```
   if n ≤ 2 then return(0)
   else
       y = ALGO(n/3)
       i = 2^n
       while i ≥ 2 do
           j = ⌊(1/2) log2 i⌋
           while j > 0 do
               i = i/2
               j = j−1
       z = ALGO(n/2)
       return (z+y)
   ```

3. Sia dato un albero binario di ricerca T, i cui nodi contengano esclusivamente una chiave intera, un puntatore al figlio sinistro e uno al figlio destro.
   Si definisca un algoritmo ricorsivo efficiente che, dati i valori interi $h_1 \geq 1$, $h_2 \geq 1$, $k_1 \geq 0$, $k_2 \geq 0$, cancelli dall'albero T tutti i nodi che, nell'albero originale fornito in ingresso, soddisfano la seguente proprietà:

   hanno chiave k pari tale che $k_1 \leq k \leq k_2$ e sono radici di sottoalberi il cui percorso esterno ha lunghezza h che soddisfa $h_1 \leq h \leq h_2$.

   Si ricorda che la *lunghezza del percorso esterno* in un albero radicato nel nodo x è la somma delle lunghezze dei percorsi da x ad una foglia.

   Nota: Non è possibile utilizzare parametri passati per riferimento né variabili globali.

4. Si supponga di voler disporre in una fila indiana n persone. Si ha a disposizione un insieme A di m affermazioni del tipo *la persona i detesta la persona j*. Se i detesta j, allora si vuole evitare di mettere i dietro a j. Definire un algoritmo che verifichi se è possibile disporre le persone in fila in modo che *nessuna persona stia davanti ad una persona che le detesta* e, in caso affermativo, produca una tale disposizione. L'algoritmo deve avere complessità $O(n + m)$.

Tema d'esame di Algoritmi e Strutture Dati I
15/07/2011

Tempo a disposizione: 3 ore.

1. Dimostrare per esteso, ricavando le costanti necessarie, le relazioni asintotiche sotto riportate:

   (a) $n \cdot log\, n - 10 \cdot n + 4 = \Omega(2 \cdot n \cdot log\, n)$

   (b) $2 \cdot n^2 + 4 \cdot n + 3 = O(n^2 - n + 7)$.

2. Si calcoli, svolgendo per esteso il ragionamento seguito, il tempo di esecuzione dell'algoritmo sotto riportato.

   1 ALGORITMO(n)
   ```
   k = 1
   t = 0
   s = 1
   while k ≤ n do
       j = k
       while j ≤ n do
           s = j + k
           j = 2 · j
       t = t + 1
       k = 2 · k
   return s + t
   ```

3. Sia dato un albero binario di ricerca T, i cui nodi contengano esclusivamente una chiave intera, un puntatore al figlio sinistro e uno al figlio destro. Sia dato, inoltre, un array A contenente possibili chiavi intere ordinate in modo crescente.
   Si definisca un algoritmo ricorsivo efficiente che cancelli dall'albero T tutti i nodi che sono a distanza $h \geq 1$ dalla radice dell'albero fornito in ingresso e che contengono una chiave presente nell'array A.

   Nota: Non è possibile utilizzare parametri passati per riferimento né variabili globali.

4. Sia dato un grafo orientato $G = (V, E)$ e un insieme $A \subseteq V$. Si definisca la distanza di $v \in V$ da A quel valore k tale che:

   - $\delta(x, v) = k$ per qualche $x \in A$ e
   - $\delta(y, v) \geq k$ per ogni $y \in A$.

   dove $\delta(x, y)$ denota l'usuale funzione di distanza tra i vertici x e y nel grafo G. Si definisca un algoritmo che, in tempo $O(|V| + |E|)$, calcoli la distanza di un vertice $v \in V$ dall'insieme A.

Tema d'esame di Algoritmi e Strutture Dati I
22/09/2011

Tempo a disposizione: 3 ore.

1. Dimostrare per esteso, ricavando le costanti necessarie, le relazioni asintotiche sotto riportate:

   (a) $2n = O(n - 3\,log_2 n)$;

   (b) $3n^3 - 6n^2 + 8n = \Theta(n^3)$.

2. Si calcoli, mostrando per esteso il procedimento seguito, il tempo di esecuzione dell'algoritmo sotto riportato.

```
1 ALGO(n)
  i = 4
  z = 0
  while i < n do
      j = i/2
      while j ≥ 1 do
          i = i + 2
          z = z + 1
          j = j - 2
      i = 2 · i
  return (z)
```

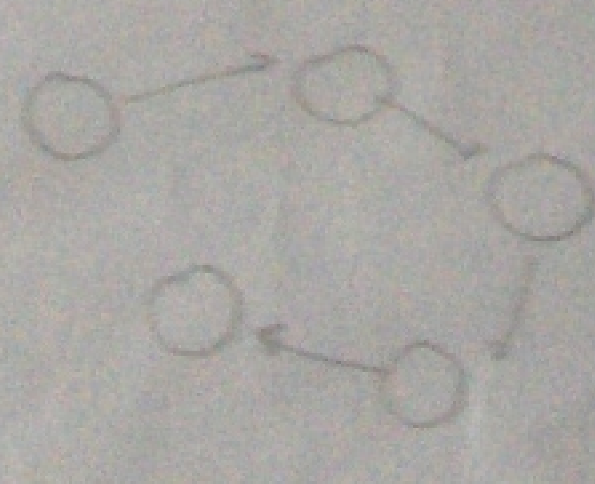
3. Sia dato un albero binario T (i cui nodi contengono esclusivamente una chiave intera, un puntatore al figlio sinistro e uno al figlio destro) che sia parzialmente ordinato, cioè tale che ogni nodo di T contiene una chiave non minore delle chiavi di entrambi i suoi figli.

   Si definisca un algoritmo ricorsivo efficiente che, dati i valori interi $k_1 \leq k_2$ e un valore intero z, cancelli dall'albero T tutti i nodi che, nell'albero originale T fornito in ingresso, soddisfano la seguente proprietà:

   hanno chiave k pari e tale che $k_1 \leq k \leq k_2$, e sono radici di sottoalberi contenenti almeno z nodi con chiave compresa tra k_1 e k_2.

   Nota: Non è possibile utilizzare parametri passati per riferimento né variabili globali.

4. Sia dato un grafo $G = (V, E)$ rappresentato tramite liste di adiacenza. Si definisca un **algoritmo iterativo** che verifichi se G è aciclico o no. In caso la risposta sia negativa, l'algoritmo deve fornire in output un possibile percorso contenente un ciclo. Il tempo impiegato dall'algoritmo deve essere $O(|V| + |E|)$.



Tema d'esame di Algoritmi e Strutture Dati I
23/02/2012

Tempo a disposizione: 3 ore.

1. Dimostrare per esteso la verità o la falsità delle seguenti affermazioni:

   (a) $n^2 = \Omega\left(n^{log_2 4/5} \cdot 5^{log_2 n}\right)$;

   (b) Si assuma che f e g siano funzioni asintoticamente crescenti e positive. Allora:

   $$log(f(n)) = \Theta(g(n)) \text{ implica } f(n) = \Theta\left(2^{g(n)}\right).$$

2. Si calcoli, mostrando per esteso il procedimento seguito, il tempo di esecuzione dell'algoritmo sotto riportato.

1 ALGO(n)

```
y = 2^n
i = n
x = log_2 y
while x > 1 do
    y = y * 2^x
    while y > 2^x do
        i = i + 2
        y = y/2
    x = x/2
return i
```

3. Si definisca un algoritmo ricorsivo efficiente che, dati un albero binario di ricerca T (i cui nodi contengono esclusivamente un campo chiave, un campo figlio sinistro e un campo figlio destro), un intero positivo $x > 0$ e un valore k, cancelli da T il nodo che soddisfa la seguente proprietà:

   contiene la più grande chiave minore di k che si trova in T a profondità non minore di x.

   Nota: non è ammesso l'utilizzo del passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Si consideri un grafo $G = (V, E)$ e un sottoinsieme $A \subseteq V$ di vertici e si definisca distanza $\delta(v, A)$ di un vertice v dall'insieme A come la minima tra le distanze del vertice v da un vertice di A. Formalmente, $\delta(v, A) = \min\{\delta(v, a) \mid a \in A\}$.

   Dato un grafo $G = (V, E)$ rappresentato tramite liste di adiacenza, un insieme A di vertici di G rappresentato come una lista puntata e due vertici u e v, si scriva lo pseudiciduce di un algoritmo che, in tempo lineare sulla dimensione di G (quindi indipendentemente dalla dimensione del sottoinsieme A) verifichi se i due vertici u e v sono alla stessa distanza da A.

Tema d'esame di Algoritmi e Strutture Dati I

11/09/2012

1. Siano f, g, h e t funzioni asintoticamente positive e crescenti. Dimostrare la verità o la falsità della seguente affermazione:

$$\text{se } \log\left(\frac{f(n)}{t(n)}\right) = \Theta(\log g(n)) \text{ e } \frac{h(n)}{f(n)} = \Theta\left(\frac{t(n)}{g(n)}\right), \text{ allora } h(n) = \Theta(t^2(n)) .$$

2. Sia dato il seguente algoritmo:

```
ALGORITMO(n)
  x = 2
  while x < n^3 do
      y = x/2
      while y < x do
          y = y + 2
          x = x + 1
      x = x * 2
  return(x)
```

Si calcoli la stima asintotica *più vicina possibile* alla funzione $T(n)$ che descrive il tempo di esecuzione dell'algoritmo.

3. Sia dato un albero binario di ricerca T, i cui nodi contengano esclusivamente una chiave intera, un puntatore al figlio sinistro e uno al figlio destro. Si definisca un algoritmo ricorsivo efficiente che, dati cinque valori interi $h_1 \geq 1$, $h_2 \geq 1$, $n_1 \geq 0$, $n_2 \geq 0$ e k, cancelli dall'albero T tutti i nodi che, nell'albero in ingresso, soddisfano la seguente proprietà:

ha chiave pari minore di k, la sua distanza dalla radice è compresa tra i valori h_1 e h_2 ed è radice di un sottoalbero i cui nodi con chiave minore di k sono in numero compreso tra n_1 e n_2 .

Nota: Non è possibile utilizzare parametri passati per riferimento nè variabili globali.

4. Sia dato un grafo orientato G. Sia $VAL[\cdot]$ un array che associa ad ogni vertice v un *valore* intero $VAL[v]$. Si definisca un algoritmo che, in tempo **lineare** sulla dimensione del grafo G, verifichi se esistono due vertici u e z tali che: $VAL[u]$ sia pari, $VAL[z]$ sia dispari e che siano tra loro reciprocamente raggiungibili.

Tema d'esame di Algoritmi e Strutture Dati I
25/06/2012

Tempo a disposizione: 3 ore.

1. (a) Dimostrare per esteso la verità o la falsità della seguente affermazione:
   siano f e g due funzioni asintoticamente crescenti e positive e j e k due valori non negativi. Allora, $f^k(n) = \Theta(2^{g^j(n)})$ implica $log_2 f(n) = \Theta(g(n))$;

   (b) Dimostrare per esteso, ricavando le costanti necessarie, la verità della seguente affermazione:

   $$n\,log_2 n - 3n - 18 = \Theta(n\,log_2 n).$$

2. Si calcoli, mostrando per esteso il procedimento seguito, il tempo di esecuzione dell'algoritmo sotto riportato.

```
1 ALGO(n)
  i = 1
  while i < n do
      j = 1/2
      while j > 0 do
          i = i + 2
          j = j - 3
      i = 2 · i
  return
```

3. Sia dato un Albero binario Parzialmente Ordinato T, in cui ogni nodo ha chiave non maggiore di quella dei suoi due figli. I nodi di T contengono esclusivamente un campo chiave, un campo figlio sinistro e un campo figlio destro. Si definisca un algoritmo ricorsivo efficiente che, dati tre valori interi strettamente positivi x, $k_1 \leq k_2$ e l'albero T, cancelli da T ogni nodo che che soddisfa la seguente proprietà:

   contiene una chiave k di valore compreso tra k_1 e k_2 ed è radice di un albero che contiene al massimo x nodi con chiave compresa tra k_1 e k_2.

   Nota: Non è possibile utilizzare parametri passati per riferimento né variabili globali.

4. Sia dato un grafo $G = (V, E)$ rappresentato tramite liste di adiacenza e due sottoinsiemi di vertici A e B. Si definisca la distanza $\delta(A, B)$ tra A e B nel seguente modo:

   $$\delta(A,B) = \begin{cases} i & \text{se per ogni } a \in A \text{ esiste un } b \in B \text{ tale che } \delta(a,b) = i \\ \infty & \text{altrimenti.} \end{cases}$$

   Si definisca un algoritmo che, dati in ingresso il grafo G, gli insiemi A e B, e un intero x, verifichi in tempo lineare sulla dimensione del grafo se $\delta(A, B) = x$.

Tema d'esame di Algoritmi e Strutture Dati I
14/02/2013

Tempo a disposizione: 3 ore.

1. A) Si dimostri la verità o la falsità (tramite controesempio) della seguente affermazione:
   Se $g(n) \cdot h(n) = \Theta(n^3)$ e $\dfrac{f(n)}{g(n)} = \Theta(n)$, allora $log(f(n)) + log(h(n)) = \Theta(log(n))$.

   B) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione:
   $$n^3\, log(n) - 13\, n = \Theta(n^2\, log(n^2)).$$

2. Sia dato il seguente algoritmo:

   1 Algoritmo(n)
   ```
   i = 0
   j = n
   while j < n^2 do
       k = 0
       x = j
       while x > i do
           i = i + 1
           x = x - 2
           k = k + 1
       i = i + 2·k
       j = j + 3·k
   return
   ```

   Calcolare, esplicitando il procedimento di soluzione seguito, il tempo di esecuzione dell'algoritmo in funzione del paramentro n.

3. Si definisca come **albero binario perfettamente bilanciato** un albero binario in cui, per ogni nodo x dell'albero, il numero di nodi nel sottoalbero sinistro di x differisce al più di uno dal numero di nodi del sottoalbero destro di x.

   Siano dato, ora, un albero binaro di ricerca T (i cui nodi contengono esclusivamente un campo chiave, un campo figlio sinistro e un campo figlio destro).

   Scrivere un algoritmo ricorsivo che, a partire da T e utilizzando opportunmente un array di appoggio A con dimensione pari al numero di nodi di T, costruisca un albero binario di ricerca perfettamete bilanciato contenente tutte e sole le chiavi di T.

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Si scriva un algoritmo che, in tempo lineare sulla dimensione del grafo orientato $G = \{V, E\}$ fornito in ingresso, costruisca **il più grande sottografo di** G contenente tutti e soli i vertici di G che possono raggiungere qualche vertice di un insieme $A \subseteq V$ ma non sono raggiungibili da nessun vertice dell'insiene $B \subseteq V$. Gli insiemi A e B sono anch'essi dati in ingresso nella forma di liste di vertici.
   Nuno

Tema d'esame di Algoritmi e Strutture Dati I
09/09/2013

Tempo a disposizione: 3 ore.

1. A) Si dimostri la verità o la falsità, esplicitando il procedimento seguito, della seguente affermazione: Se $f(n) = \Theta(\sqrt{g(n)})$ e $log\, g(n) = \Theta(2^n)$; allora $log\, f(n) = \Theta(log\, n)$.

   B) Si dimostri o si confuti la seguente affermazione, esplicitando il procedimento seguito e le costanti necessarie, se esse esistono: $n^2 - n + \frac{1}{n} = \Theta(n^2)$.

2. Calcolare, **esplicitando il procedimento completo di soluzione seguito**, il tempo di esecuzione del seguente algoritmo in funzione del paramentro n.

1 Algoritmo(n)

```
x = 2^n
while x > 2 do
    y = x/2
    z = 2
    while x > y do
        x = x - z
        z = 2 · z
return
```

3. Sia dato un albero binaro T (i cui nodi contengono esclusivamente un campo chiave intero, un campo figlio sinistro e un campo figlio destro).

   Scrivere un algoritmo ricorsivo efficiente che elimini da T tutti i nodi che contengono una chiave pari e, contemporaneamente, costruisca un albero binario di ricerca T' contenente tutti i nodi eliminati da T. L'algoritmo richiesto deve restituire l'albero T' e non può avere T' tra i suoi parametri di ingresso.

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Sia dato grafo orientato $G = \langle V, E \rangle$, rappresentato con liste di adiacenza e un array A contenente un sottoinsieme dei vertici di G (quindi, $A \subseteq V$). Si scriva un algoritmo che, in tempo lineare sulla dimensione di G, verifichi se esiste un percorso di G della forma $v_1 \to v_2 \to \cdots \to v_k \to v_1$ tale che $A \subseteq \{v_1, \cdots, v_k\}$.

Tema d'esame di Algoritmi e Strutture Dati I
26/06/2014

Tempo a disposizione: 3 ore.

1. Si dimostri per esteso la verità o la falsità di ciascuna delle seguenti affermazioni:

   A) Si dimostri, esplicitando per esteso il procedimento seguito, la verità o la falsità della seguente affermazione: Se $f(n) = \Theta(log_2\, g(n))$ e $h(n) = \Theta(2^n)$, allora $log_2\, h(f(n)) = \Theta(log_2\, g(n))$. Si assuma che le fuzioni f e g siano asintoticamente crescenti e positive.

   B) Calcolare, se estono, le costanti necessarie a dimostrare la seguente affermazione: $e^{n-1} - 2^n = \Theta(e^n)$ (si espliciti per esteso il procedimento seguito).

2. Sia dato il seguente algoritmo:

   1 Algoritmo(n)
   ```
   x = 2^n
   y = 0
   while x > 2 do
       z = y + 1
       y = log2 x
       while y > 0 do
           y = y - 2
           z = z * 2
       x = x/z
   return
   ```

   Calcolare, **esplicitando tutto il procedimento di soluzione seguito**, la stima asintotica più vicina possibile al tempo di esecuzione $T(n)$ dell'algoritmo.

3. Sia dato un Albero Binario di Ricerca T, i cui nodi contengono i seguenti **4 campi**: (i) un campo chiave di tipo intero; (ii) due campi uno per figlio sinistro e uno per figlio destro; e (iii) un campo addizionale di tipo intero, che associa ad ogni nodo *il numero di foglie contenute nel sottoalbero di cui esso è radice*. Siano, inoltre, dati due interi k_1 e k_2 (con $k_1 \leq k_2$) e due interi $x > 0$ e $h > 0$. Definire un algoritmo ricorsivo efficiente che cancelli da T tutti nodi che, nell'albero iniziale T in ingresso, soddisfano la seguente condizione:
   "hanno chiave pari compresa tra k_1 e k_2, hanno profondità minore o uguale ad h e il percorso esterno del sottoalbero di cui sono radici è maggiore o uguale a x."
   L'albero risultante deve essere dello stesso tipo dell'albero in ingresso (si faccia, quindi, attenzione a calcolare correttamente i valori del quarto campo dei nodi).
   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Sia dato un arbitrario grafo orientato $G = (V, E)$ e un vertice $v \in V$. Un grafo $G' = (V, E')$ di G si dice "sottografo massimale di G che ammette almeno un ordinamento topologico che parte da v" se: (i) $E' \subseteq E$ e (ii) per ogni $(u, u') \in E \setminus E'$, il grafo $(V, E' \cup \{(u, u')\})$ non ammette un ordinamento topologico che parte da v.

   Si scriva un algoritmo che, in tempo lineare sulla dimensione di G, costruisca un sottografo massimale G' di G che ammette almeno un ordinamento topologico che parte da v (il grafo G' risultante deve essere un nuovo grafo, cioè un grafo distinto da G).

Tema d'esame di Algoritmi e Strutture Dati I
18/07/2014

Tempo a disposizione: 3 ore.

1. A) Si dimostri la verità o la falsità (tramite controesempio) della seguente affermazione: Se $z(n) = \Theta(log\, n^3)$ e $g(n) = \Theta(log(log\, n)^3)$, allora $log\, z(n) = \Theta(2^{g(n)})$.
   B) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione: $log\left(\frac{n}{10}\right) = \Theta(log\, n^5)$

2. Sia dato il seguente algoritmo:

1 Algoritmo(n)
```
x = n
j = x/2
while x > 1 do
    z = j
    while j > 1 do
        j = j/4
        x = x - 2
    x = 2^(2z) / 2^x
    j = x/2
return
```

Calcolare, esplicitando il procedimento di soluzione seguito, il tempo di esecuzione dell'algoritmo in funzione del paramentro n.

3. Sia dato un albero binaro parzialmente ordinato T (in cui i figli di ogni nodo hanno chiave non maggiore di quella del nodo stesso).

   A) scrivere un algoritmo ricorsivo efficiente che, dato in ingresso T e un valore $k > 0$, restituisca la **lista ordinata delle chiavi pari e minori di** k contenute nell'albero (l'albero T non deve essere necessariamente preservato).

   B) come modifichereste la procedura del punto A) se venisse richiesto di risolvere il problema senza modificare l'albero originario T?

Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Si definisca un *albero orientato* come un grafo nel quale: (i) esiste un unico suo vertice (detto la radice) che raggiunge ogni altro suo vertice tramite un qualche percorso orientato; (ii) per ogni vertice v del grafo, esiste un solo percorso orientato che raggiunge v dalla radice. Una *foresta orientata* è, dunque, un insieme di alberi orientati che abbia cardinalità ≥ 1.

   Si scriva un algoritmo che, in tempo lineare sulla dimensione del grafo orientato $G = \{V, E\}$ fornito in ingresso, verifichi se il grafo in ingresso è una foresta orientata oppure no.

Tema d'esame di Algoritmi e Strutture Dati I
17/09/2014

Tempo a disposizione: 3 ore.

1. Per i seguenti esercizi si riporti per esteso il procedimento di soluzione seguito:

   A) Si dimostri la verità o la falsità (tramite controesempio) della seguente affermazione: Se $z(n) = \Theta(n^5)$ e $g(n) = \Theta(log(log\, n)^2)$, allora $log\, z(n) = \Theta(2^{g(n)})$;

   B) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione: $2\,n\,log\,n + n - 8\,log\,n = \Theta(n\,log\,n)$.

2. Sia dato il seguente algoritmo:

```
1 Algoritmo(n)
  x = 2^n
  while x > 2 do
      y = 0
      while log x > y do
          y = y + 1
          x = x/2
  z = n/x
  return Algoritmo(z)
```

   Calcolare, esplicitando il procedimento di soluzione seguito, il tempo di esecuzione dell'algoritmo in funzione del paramentro n.

3. Sia dato un Albero Binario di Ricerca T a chiavi intere (in cui i campi di ogni nodo prevedono sol la chiave e i puntatori ai due figli) e due valori interi k_1 e k_2. Scrivere un algoritmo **iterativo** che cancelli da T tutti i nodi che abbiano chiave k tale che $k_1 \leq k \leq k_2$.

   È ammesso l'uso di chiamate di procedura ma non di chiamate ricorsive. Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Un grafo orientato $G = \langle V, E\rangle$ si dice **completo** se ogni suo vertice è adiacente ad ogni altro vertice Si ricorda, inoltre, che se $V' \subseteq V$, il sottografo di G **indotto** da V' è il massimo sottografo di G che ha V' come insieme dei vertici.

   Siano dati in ingresso un grafo orientato $G = \langle V, E\rangle$ **rappresentato come matrice di adiacenza** una sequenza di vertici distinti $V' \subseteq V$ (contenuta in un array di lunghezza $|V'|$).

   A) si scriva un algoritmo che verifichi, in tempo $O(|V'|^2)$, se il sottografo di G indotto da V' completo oppure no;

   B) si scriva un algoritmo che in tempo $O(|V'|^3)$ calcoli la più lunga **sottostringa** V'' di V' (cio la massima sottosequenza di elementi contigui nella sequenza V') tale che il sottografo di indotto da V'' è completo.

Tema d'esame di Algoritmi e Strutture Dati I
29/01/2015

Tempo a disposizione: 3 ore.

1. A) Si dimostri per esteso la verità o la falsità (tramite controesempio) della seguente affermazione:

$$\text{Se } g(n) \cdot h(n) = \Theta(n^3) \text{ e } \frac{f(n)}{g(n)} = \Theta(n), \text{ allora } log(f(n)) + log(h(n)) = \Theta(log(n)).$$

Si assuma che tutte le fuzioni siano asintoticamente crescenti e positive.

B) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione:

$$n^2\, log(n^2) + 15\, n^2 = \Theta(n^2\, log(n)).$$

Si fornisca **per esteso** il procedimento di dimostrazione seguito.

2. Sia dato il seguente algoritmo:

1 Algoritmo(n)

```
x = n
while x > 2 do
    z = x
    j = z/2
    while j > 1 do
        j = j/2
        x = x - 1/2
    x = 2^(z-x+1/2)
return
```

Calcolare, esplicitando il procedimento di soluzione seguito, la stima asintotica più vicina possibile al tempo di esecuzione $T(n)$ dell'algoritmo.

3. Sia dato un albero binario di ricerca T (i cui nodi contengono esclusivamente un campo chiave di tipo intero, un campo figlio sinistro e un campo figlio destro) e due interi k_1 e k_2 (con $k_1 \leq k_2$). Definire un algoritmo ricorsivo efficiente che cancelli da T tutti e soli i nodi a profondità massima con chiave pari compresa tra k_1 e k_2. Più precisamente, l'algoritmo deve cancellare tutti e soli i nodi che soddisfano **entrambe** le seguenti condizioni:

   (a) hanno chiave pari e compresa tra k_1 e k_2;

   (b) non hanno alcun discendente che soddisfi la condizione (a).

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Diciamo che due vertici v e u di un grafo orientato $G = \langle V, E \rangle$ sono *separati* da un insieme $X \subseteq V$, se ogni possibile percorso da v a u passa necessariamente per un vertice di X e, viceversa, ogni percorso da u a v passa necessariamente per un vertice di X. Si scriva un algoritmo che, in **tempo lineare sulla dimensione del grafo** orientato G, fornito in ingresso e rappresentato mediante liste di adiacenza, verifichi se due vertici in ingresso u e v sono separati dall'insieme di vertici $X \subseteq V$, anch'esso dato in ingresso come array di vertici.

Tema d'esame di Algoritmi e Strutture Dati I
20/02/2015

Tempo a disposizione: 3 ore.

1. A) Si dimostri per esteso la verità o la falsità (tramite controesempio) della seguente affermazione: Se $z(n) = \Theta(2^{g(n)})$ e $h(n) = \Theta(log\, g(n)$, allora $log\, z(n) = \Theta(2^{h(n)})$.
   Si assuma che le fuzioni g, z e f siano asintoticamente crescenti e positive.

   B) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione: $5\,n^2 - 8\,\sqrt{n} + 1 = \Theta(n^2)$. Il procedimento di soluzione deve essere riportato per esteso.

2. Sia dato il seguente algoritmo:

   1 Algoritmo(n)
   $x = n$
   while $x > 4$ **do**
       $z = 3 * \sqrt{x}$
       $j = 1$
       while $j > \sqrt{x}$ **do**
           $j = j * 2$
           $z = z - j$
       $x = \cancel{\frac{x}{(z+1)}}$ $\frac{x}{z-2}$
   return

   Calcolare il tempo di esecuzione dell'algoritmo in funzione del paramentro n, esplicitando il procedimento di soluzione seguito.

3. Sia dato un albero binaro di ricerca T (i cui nodi contengono esclusivamente un campo chiave intero, un campo figlio sinistro e un campo figlio destro).

   Scrivere un algoritmo **iterativo** efficiente che costruisca una lista ordinata (in ordine crescente), contenente tutti i nodi di T le cui chiavi siano contenute nell'intervallo di interi $[k_1, k_2]$ e che si trovino a profondità comprese nell'intervallo di interi $[h_1, h_2]$ (dove T, k_1, k_2, h_1 e h_2 sono tutti parametri di input dell'algoritmo).

   Non è ammesso l'uso di passaggio di parametri per riferimento né l'impiego di variabili globali.

4. Sia dato grafo orientato $G = \{V, E\}$, rappresentato con liste di adiacenza, un insieme di nodi $X \subseteq V$ memorizzato in un array e un intero non negativo k.

   Si scriva un algoritmo che, in tempo lineare sulla dimensione di G, raccolga in una lista L i nodi di G a partire dai quali **tutti i percorsi** che raggiungono un nodo di X hanno lunghezza maggiore di k.

Tema d'esame di Algoritmi e Strutture Dati I
25/06/2015

Tempo a disposizione: 3 ore.

1. Si svolgano i seguenti esercizi:

   (a) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione: $log\left(\frac{2^n}{n}\right) - 1 = \Theta(log(2^{2n}))$.

   (b) Si dimostri la verità o la falsità (tramite controesempio) della seguente affermazione: Se $z(n) = \Theta\left(2^{g(n)}\right)$ e $h(n) = \Theta\left(log\,log\,g(n)\right)$, allora $log\,log\,z(n) = \Theta\left(2^{h(n)}\right)$.
   Si assuma che le fuzioni g, z e f siano asintoticamente crescenti e positive.

2. Si consideri il seguente algoritmo:

   1 Algoritmo(n)

   ```
   x = n
   y = x/3
   while x > y + 1 do
       y = y/3
       while y > 0 do
           y = y - 1
           x = x - 3
       y = x/3
   return
   ```

   Si individui la stima asintotica più vicina possibile al tempo di esecuzione dell'algoritmo in funzione del parametro n.

3. Sia dato un albero binario di ricerca T, i cui nodi contengano esclusivamente una chiave intera, un puntatore al figlio sinistro e uno al figlio destro.
   Si definisca un **algoritmo iterativo** efficiente che, dato il rifetimento all'albero T e un valore intero k, restituisca il riferimento al nodo di T contenente la più piccola chiave strettamente maggiore di k.

4. Sia dato un grafo orientato G e un array $VAL[\cdot]$ che associa a ogni vertice v un *numero naturale* $VAL[v]$. Si definisca un algoritmo che, in tempo lineare sulla dimensione del grafo G, verifichi se esistono due vertici della stessa parità (cioè i cui numeri naturali associati siano o entrambi pari o entrambi dispari) tali che entrambe le seguenti condizioni valgano:

   (a) nel grafo esiste almeno un ciclo che contiene sia il primo che il secondo vertice;

   (b) ogni percorso che connette i due vertici passa solo per vertici della loro stessa parità.

Tema d'esame di Algoritmi e Strutture Dati I
11/09/2015

Tempo a disposizione: 3 ore. 14.30 17.30

1. Si svolgano i seguenti esercizi, esplicitando lo svolgimento completo:

   (a) Si individuino, se esistono, le costanti necessarie a dimostrare la seguente relazione:

   $$5n^2 - n + \log n = \Theta\left(n^2\right)$$

   (b) Si dimostri la verità o la falsità (tramite controesempio) della seguente affermazione:

   se $h(n) = \Theta(2^n)$ e $log_2\, h(f(n)) = \Theta(log_2\, g(n))$, allora $f(n) = \Theta(log_2\, g(n))$.

   Si assuma che le fuzioni g e f siano asintoticamente crescenti e positive.

2. Si consideri il seguente algoritmo:

   1 Algoritmo(n)
   ```
   x = 2^n
   y = x^2
   while x > 2 do
       z = 1
       while y > z do
           y = y/2
           z = z * 2
       x = √y
   return
   ```

   Si individui, esplicitando lo svolgimento completo, la stima asintotica più vicina possibile al tempo di esecuzione dell'algoritmo in funzione del parametro n.

3. Sia dato un albero binario di ricerca T, i cui nodi contengano esclusivamente una chiave intera, un puntatore al figlio sinistro e uno al figlio destro.
   Si definisca un **algoritmo iterativo** efficiente che, dato il riferimento all'albero T e un valore intero k, cancelli da T il nodo contenente la **più piccola chiave pari maggiore di** k.

4. Siano dato un grafo orientato $G = (V, E)$, e una permutazione π (in forma di array) di un qualche sottoinsieme $V' \subseteq V$. Si definisca un algoritmo che, in tempo lineare sulla dimensione del grafo G, verifichi se π è un ordinamento topologico del sottografo di G indotto da V', senza costruire il sottografo indotto.