

Algoritmi e Strutture Dati

27 Giugno 2019

Cognome Nome Matricola

Note

1. La leggibilità è un prerequisito: parti difficili da leggere potranno essere ignorate.
2. Quando si presenta un algoritmo è fondamentale spiegare l'idea sottostante e motivarne la correttezza.
3. L'efficienza e l'aderenza alla traccia sono criteri di valutazione delle soluzioni proposte.
4. Si consegnano tutti i fogli, con nome, cognome, matricola e l'indicazione *bella copia* o *brutta copia*.

Domande

Domanda A (5 punti) Si determini la soluzione asintotica della seguente equazione di ricorrenza:

$$T(n) = 4T(n/3) + n^2 + 1$$

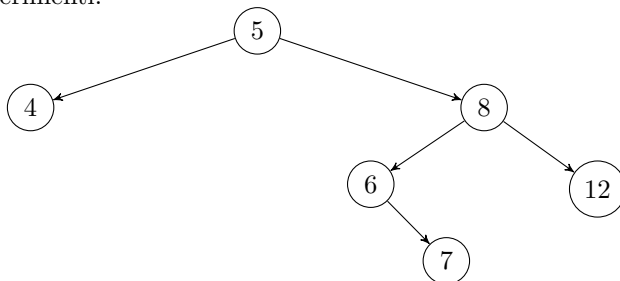
Soluzione: Rispetto allo schema generale si ha $a = 4$, $b = 3$, $f(n) = n^2 + 1$. Si osserva che $\log_b a = \log_3 4 < 2$ quindi $f(n) = \Omega(n^{\log_b a + \epsilon})$ (per $0 < \epsilon \leq 2 - \log_3 4$). In aggiunta vale la condizione di regolarità, ovvero $af(\frac{n}{b}) \leq cf(n)$ per qualche $c < 1$. Infatti $af(\frac{n}{b}) = 4f(\frac{n}{3}) = 4(\frac{n^2}{9} + 1) \leq cn^2 \leq c(n^2 + 1)$, asintoticamente, quando $4/9 < c < 1$. Più precisamente, per avere

$$4(\frac{n^2}{9} + 1) \leq cn^2$$

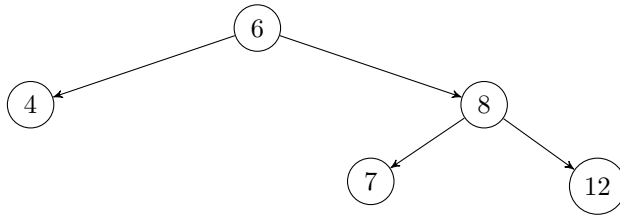
Occorre che $c \geq \frac{4}{9} + \frac{1}{n^2}$, che, per $n \geq 2$ si ottiene quando $c \geq \frac{4}{9} + \frac{1}{4} = \frac{25}{36} < 1$.

Domanda B (5 punti) Dare la definizione di albero binario di ricerca. Specificare l'albero ottenuto inserendo, con la procedura vista a lezione, a partire da un albero vuoto, i nodi aventi le seguenti chiavi: 5, 4, 8, 6, 12, 7. Si supponga che dall'albero così ottenuto si cancelli il nodo con chiave 5 e si indichi l'albero ottenuto. Sia per gli inserimenti che per la cancellazione, motivare sinteticamente il risultato ottenuto.

Soluzione: Per la definizione di albero binario di ricerca e la descrizione delle procedure di inserimento e cancellazione si consulti il libro. Sullo specifico esempio si ottiene, dopo gli inserimenti:



La cancellazione di 5 quindi produce:



Domanda C (4 punti) Si consideri un insieme di 6 attività $a_i, 1 \leq i \leq 6$, caratterizzate dai seguenti vettori \mathbf{s} e \mathbf{f} di tempi di inizio e fine:

$$\mathbf{s} = (2, 3, 5, 2, 1, 9)$$

$$\mathbf{f} = (4, 7, 8, 9, 10, 11).$$

Determinare l'insieme di massima cardinalità di attività mutuamente compatibili selezionato dall'algoritmo greedy GREEDY_SEL visto in classe. Motivare il risultato ottenuto descrivendo brevemente l'algoritmo.

Soluzione: Si considerano le attività ordinate per tempo di fine, e ad ogni passo si sceglie l'attività che termina prima, rimuovendo quelle incompatibili. Si ottiene così l'insieme di attività: $\{a_1, a_3, a_6\}$

Esercizi

Esercizio 1 (8 punti) Realizzare una funzione booleana `triple(A)` che dato un array di numeri $A[1..n]$ verifica se esistono tre indici (non necessariamente diversi) i, j e k , tali che $A[i] + A[j] = A[k]$ e restituisce un corrispondente valore booleano. Ad esempio per $[85, 10, 29, 93, 95]$ restituisce true, dato che $85 + 10 = 95$, mentre per $[70, 45, 83, 21, 68]$ restituisce false. Si suggerisce di ordinare l'array. Valutare la complessità.

Soluzione: Assumiamo inizialmente che l'array A sia ordinato. In questo caso, si può cercare, per ogni indice k , una coppia (i, j) tale che $A[k] = A[i] + A[j]$.

assume $A[1..n]$ ordinato

cerca tre indici i, j, k tali che $A[i] + A[j] = A[k]$

```

Triple(A)
  n = A.length
  found = false
  k=1
  while (not found) and (k <= n)
    # cerca se ci sono due indici tali che  $A[i] + A[j] = A[k]$ 
    i=1
    j=n
    # invariante:
    #   per ogni  $i' < i$ , e  $j'$  qualunque  $A[i'] + A[j'] < A[k]$ 
    #   per ogni  $j' > j$ , e  $i'$  qualunque  $A[i'] + A[j'] > A[k]$ 
    while (i <= j) and ( $A[i] + A[j] <> A[k]$ )
      if  $A[i] + A[j] < A[k]$ 
        i = i+1
      else
        j = j-1
    found = i <= j

```

k=k+1

return found

La complessità è $O(n^2)$: infatti, il ciclo interno, ad ogni iterazione incrementa i o decrementa j , quindi il numero massimo di iterazioni è n , e viene ripetuto al più n volte per effetto del ciclo esterno.

Se l'array non è ordinato, si può ordinarlo preventivamente, cosa che si può fare con costo $O(n \log n)$ e quindi senza alterare la complessità asintotica complessiva.

Esercizio 2 (8 punti) Data una stringa di numeri interi $A = (a_1, a_2, \dots, a_n)$, si consideri la seguente ricorrenza $z(i, j)$ definita per ogni coppia di valori (i, j) con $1 \leq i, j \leq n$:

$$z(i, j) = \begin{cases} a_j & \text{if } i = 1, 1 \leq j \leq n, \\ a_{n+1-i} & \text{if } j = n, 1 < i \leq n, \\ (z(i-1, j) \cdot z(i, j+1)) - z(i-1, j+1) & \text{altrimenti.} \end{cases}$$

1. Si fornisca il codice di un algoritmo iterativo bottom-up $\text{COMPUTE_Z}(A)$ che, data in input la stringa A restituisca in uscita il valore $z(n, 1)$.
2. Si valuti il numero esatto $T_{CZ}(n)$ di operazioni tra interi eseguite dall'algoritmo sviluppato al punto (a).

Soluzione:

1. Date le dipendenze tra gli indici nella ricorrenza, un modo corretto di riempire la tabella è attraverso una scansione “reverse column-major”, in cui calcoliamo gli elementi della tabella in ordine decrescente di indice di colonna e, all'interno della stessa colonna, in ordine crescente di indice di riga. Il codice è il seguente.

```
COMPUTE_Z(A)
n = length(A)
for i=1 to n do
    z[1,i] = a_i
    z[i,n] = a_{n+1-i}
for j=n-1 downto 1 do
    for i=2 to n do
        z[i,j] = (z[i-1,j] * z[i,j+1]) - z[i-1,j+1]
return z[n,1]
```

Si osservi che un altro modo corretto di riempire la tabella è attraverso una scansione “reverse diagonal”, che scansiona per diagonali parallele alla diagonale principale partendo da quella contenente solo $z[1, n]$.

2. Ogni iterazione del doppio ciclo dell'algoritmo esegue due operazioni tra interi, e quindi

$$\begin{aligned} T_{CZ}(n) &= \sum_{j=1}^{n-1} \sum_{i=2}^n 2 \\ &= \sum_{j=1}^{n-1} 2(n-1) \\ &= 2(n-1)^2. \end{aligned}$$

Equivalentemente, basta osservare che l'algoritmo esegue due operazioni per ogni elemento di una tabella $(n - 1) \times (n - 1)$.

Nota: Correzione, risultati e visione dei compiti: *Giovedì 4 Luglio, ore 9:30 2BC/30*