

Laurea in “Informatica”

Corso di “Algoritmi e Strutture di Dati”

21 Gennaio 2014

1. Tempo disponibile 180 minuti.
2. Non è possibile consultare appunti, libri o persone, né uscire dall'aula.
3. Le soluzioni degli esercizi devono:
 - spiegare a parole l'algoritmo usato (anche con l'aiuto di eventuali disegni)
 - fornire e commentare lo pseudocodice (dettagliando a parole il significato delle variabili)
 - giustificare la correttezza e la complessità (con tutti i passaggi matematici necessari)
4. Un esercizio può ammettere più soluzioni: a soluzioni computazionalmente più efficienti e/o concettualmente più semplici sono assegnati punteggi maggiori

1. Si calcoli la complessità $T(n)$ della seguente procedura ricorsiva, con $n > 0$, specificando se tale complessità è polinomiale o esponenziale nella *dimensione* dell'input:

```
integer mystery(integer n)
  integer k ← - 1
  for integer i ← n - 1 to n + 1 do
    k ← k + 1
  if n > 7 then
    return 3 · mystery( $\lfloor n/k \rfloor$ ) + mystery( $\lfloor n/2 \rfloor$ ) +  $\lfloor n/k \rfloor$ 
  else
    return n
```

2. Data una lista L di interi, si vuole modificarla cancellando tutti gli elementi adiacenti che hanno valori consecutivi, mantenendo l'ordine iniziale degli elementi (p.e. se l'ingresso è $L = \underline{4}, 5, \underline{6}, 2, 7, \underline{3}, \underline{2}, \underline{1}, \underline{2}, 5$ allora il risultato è $L = 2, 7, 5$). Si scriva lo pseudo-codice di un algoritmo efficiente *utilizzando gli operatori* per le liste.

3. Si scriva lo pseudo-codice di un algoritmo che, dati un intero k ed un albero binario T i cui nodi contengono valori interi, cancelli in tempo *ottimo* ogni *foglia* di livello k contenente un intero uguale a quello contenuto nel *padre*, assumendo T realizzato con *puntatori*.

4. Si descriva la struttura di dati *dizionario*, con le operazioni tipicamente utilizzate, e la sua realizzazione efficiente tramite *tabella hash*. Si discuta in quali casi la realizzazione con tabella hash è da preferirsi rispetto ad altre realizzazioni.

5. Si scriva lo pseudo-codice dell'algoritmo di Kruskal per calcolare il minimo albero di copertura di un grafo non orientato pesato, dimostrandone correttezza e complessità. Successivamente, si esegua a mano l'algoritmo sul grafo $G = (N, A)$, con $N = \{1, 2, 3, 4, 5\}$, $A = \{[1,3], [1,4], [1,5], [2,3], [3,4], [4,5]\}$ e pesi $w[1,3]=1$, $w[1,4]=3$, $w[1,5]=4$, $w[2,3]=6$, $w[3,4]=5$, $w[4,5]=2$, mostrando ad ogni passo il contenuto delle strutture di dati usate.

6. Dato un insieme A , di n interi distinti, si vuole decidere se esistono due sottoinsiemi *non vuoti e disgiunti* S e T di A tali che il triplo del prodotto degli elementi in S sia uguale al doppio della somma degli elementi in T . Scrivere lo pseudo-codice di un algoritmo *non deterministico* che richieda tempo polinomiale.