Algoritmi per l'Ingegneria – Ingegneria dell'Informazione

Compito Tipo (Durata: 2h30m)

Nome, Cognome, Matricola: ______________________________

Prima Parte: domande a risposta unica

Si forniscano negli appositi spazi le risposte alle seguenti domande. Ogni risposta esatta vale 2 punti. E' necessario conseguire almeno 6 punti per essere ammessi alla correzione della seconda parte. **ATTENZIONE: gli studenti devono riportare sul foglio di bella <u>brevi</u> procedimenti che conducono alle risposte.**

1. Sia $\Pi = \mathcal{I} \times \mathcal{S}$ il problema della risoluzione di equazioni di primo grado $ax + b = 0$, con a e b coefficienti reali **positivi** (ovvero, $a, b \in \boldsymbol{R}^+$). Si definiscano gli insiemi $\mathcal{I}$ e $\mathcal{S}$.

   $\boxed{\mathcal{I} = \ldots\ldots\ldots, \mathcal{S} = \ldots\ldots}$

2. Si determini l'ordine di grandezza stretto (Θ) della seguente ricorrenza utilizzando il Master Theorem: $T(n) = 4T(n/16) + \sqrt{n}$.

   $\boxed{T(n) = \ldots\ldots\ldots\ldots}$

3. Sia $\boldsymbol{y} = (8, 0, 0, 0, 16, 0, 0, 0)$ e sia $\boldsymbol{x} = DFT_8^{-1}(\boldsymbol{y})$. Allora

   $\boxed{\boldsymbol{x} = (\ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots, \ldots)}$

4. Sia $\boldsymbol{p} = (1, 10, 5, 4)$ un vettore di dimensioni. Il minimo numero di prodotti tra scalari necessario a calcolare il prodotto di tre matrici A_i di dimensioni $p_{i-1} \times p_i$, $1 \leq i \leq 3$ è:

   $\boxed{m(1,3) = \ldots\ldots}$

5. Si calcoli il numero esatto $T(n)$ di esecuzioni del comando $a \leftarrow a + 1$ nel seguente frammento di codice:

```
for i ← 1 to n − 1 do
    for j ← i to i + 1 do
        for k ← j to 2j − 1 do
            a ← a + 1
```

   $\boxed{T(n) = \ldots\ldots}$

Seconda Parte: risoluzione di problemi

Si forniscano soluzioni esaurienti, rigorose e corredate di prova di correttezza ai due problemi seguenti. Gli algoritmi da vanno descritti utilizzando lo **pseudocodice** usato in classe.

Esercizio 1 ([11 Punti]) Sia n una potenza di due. Per un dato problema computazionale Π, si supponga di avere un algoritmo iterativo A_1 la cui complessità su istanze di taglia n sia $T_1(n) = n^2$, e un algoritmo ricorsivo A_2 la cui complessità sia regolata dalla seguente ricorrenza:

$$T(n) = \begin{cases} 1 & n = 1, \\ 2T(n/2) + 7n & n > 1. \end{cases}$$

1. ([7 Punti]) Analizzando l'albero delle chiamate, si determini la complessità $T_{n_0}(n)$ del generico algoritmo ibrido A_H che utilizza la strategia ricorsiva di A_2 per taglie dell'istanza $n > n_0$ e invoca A_1 per taglie $n \leq n_0$, dove n_0 è una potenza di due.

2. ([4 Punti]) Si determini il valore $\bar{n}_0$ che fornisce il miglior algoritmo ibrido per $n > \bar{n}_0$. (*Attenzione:* si ricordi che $\frac{d}{dx} \log_2 x = \frac{\log_2 e}{x}$)

Esercizio 2 [12 Punti] Per $n > 1$, dato un array bidimensionale $A[1..n, 1..n]$ di interi e una coppia di indici $[s, t]$ con $1 \leq s, t \leq n$, si definisce $[s, t]$-*cammino in* A una sequenza $\Pi_{[s,t]} = \langle A[i_0, j_0], A[i_1, j_1], \ldots, A[i_m, j_m] \rangle$ con $1 = i_0 \leq i_1 \leq \ldots \leq i_m = s$, $1 = j_0 \leq j_1 \leq \ldots \leq j_m = t$ e tale che per $0 \leq k < m$, $(i_{k+1} - i_k) + (j_{k+1} - j_k) = 1$. In parole, un $[s, t]$-cammino è una sequenza di elementi dell'array a partire da $[1, 1]$ sino a $[s, t]$, con il vincolo che elementi successivi nel cammino si trovano o sulla stessa riga e in colonne successive oppure sulla stessa colonna e in righe successive. Il *costo* di $\Pi_{[s,t]}$ è dato da $\sum_{k=0}^{m} A[i_k, j_k]$. Dato in ingresso un array $A[1..n, 1..n]$ di interi positivi, si consideri ora il problema di determinare il costo minimo di un $[n, n]$-cammino in A.

1. ([7 Punti]) Si enunci e si dimostri una proprietà di sottostruttura ottima per il problema e si derivi una ricorrenza dalla proprietà enunciata.

2. ([5 Punti]) Si fornisca lo pseudocodice del risultante algoritmo **iterativo** di programmazione dinamica. Per avere punteggio pieno, l'algoritmo deve eseguire $O(n^2)$ operazioni aritmetiche tra interi.