Algoritmi e Strutture Dati
10 Luglio 2020

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 e motivarne la correttezza.
3. L'efficienza e l'aderenza alla traccia sono criteri di valutazione delle soluzioni proposte.
4. Si consegna la scansione dei fogli di bella copia, e come ultima pagina un documento di identità.

Domande

Domanda A (8 punti) Definire formalmente la classe $\Theta(g(n))$. Dimostrare le seguenti affermazioni o fornire un controesempio:

i. se $f(n), f'(n) \in \Theta(g(n))$ allora $f(n) + f'(n) \in \Theta(g(n))$;

ii. $f(n), f'(n) \in \Theta(g(n))$ allora $f(n) * f'(n) \in \Theta(g(n))$;

Domanda B (6 punti) Si consideri un insieme di 7 attività $a_i, 1 \leq i \leq 7$, caratterizzate dai seguenti vettori **s** e **f** di tempi di inizio e fine:

$$\mathbf{s} = (1, 4, 2, 3, 7, 8, 11) \qquad \mathbf{f} = (3, 6, 9, 10, 11, 12, 13).$$

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.

Esercizi

Esercizio 1 (9 punti) Realizzare una funzione booleana `checkSum(A,B,C,n)` che dati tre array `A[1..n]`, `B[1..n]` e `C[1..n]` con valori numerici, verifica se esistono tre indici i, j, k con $1 \leq i, j, k \leq n$ tali che $A[i] + B[j] = C[k]$. Valutarne la complessità.

Esercizio 2 (9 punti) Data una stringa di numeri interi $A = (a_1, a_2, \ldots, a_n)$, si consideri la seguente ricorrenza $c(i, j)$ definita per ogni coppia di valori (i, j) con $1 \leq i, j \leq n$:

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

1. Si fornisca il codice di un algoritmo iterativo bottom-up COMPUTE_C(A) che, data in input la stringa A restituisca in uscita il valore $c(n, 1)$.
2. Si valuti il numero esatto $T_{CC}(n)$ di moltiplicazioni tra interi eseguite dall'algoritmo sviluppato al punto (1).