

Algoritmi e Strutture Dati – Parte A – 24/07/2020

Esercizio A1, punti ≥ 8

Trovare i limiti superiori e inferiori più stretti possibili per la seguente famiglia di equazioni di ricorrenza, per valori di a interi positivi.

$$T(n) = \begin{cases} aT(\lfloor n/2 \rfloor) + n^{a-1} & n \geq 2 \\ 1 & n < 2 \end{cases}$$

Esercizio A2 – Punti ≥ 10

Scrivere un algoritmo

boolean closeDuplicates(**int**[] A , **int** n , **int** k)

che prenda in input un vettore A contenente n interi e restituisca **true** se e solo se esiste un valore che è ripetuto almeno due volte nel vettore a distanza inferiore o uguale a $k < n$:

$$\exists i, j : 1 \leq i < j \leq n \wedge j - i \leq k \wedge A[i] = A[j]$$

Discutere informalmente la correttezza dell'algoritmo e calcolare la sua complessità computazionale.

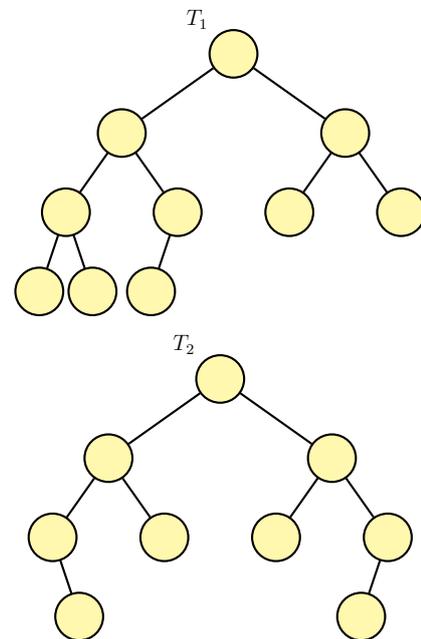
- Dato il vettore $A = [1, 2, \mathbf{3}, 5, \mathbf{3}, 7, 1]$ e il valore $k = 2$, l'algoritmo deve restituire **true** perché il valore 3 è ripetuto due volte a distanza 2 (valori evidenziati).
- Dato il vettore $A = [1, 2, \mathbf{3}, 5, 7, \mathbf{3}, 1]$ e il valore $k = 2$, l'algoritmo deve restituire **false** perché il valore 3 è ripetuto due volte a distanza 3 (valori evidenziati) mentre il valore 1 è ripetuto a distanza 6.
- Ovviamente, dato il vettore $A = [1, 2, 3, 4, 5, 6]$ e qualunque valore k , l'algoritmo deve restituire **false** perché non ci sono valori ripetuti.

Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Esercizio A3, punti ≥ 12

Scrivere un algoritmo **boolean** isSymmetric(TREE T) che prenda in input un albero binario T non vuoto e restituisca **true** se T è simmetrico, **false** altrimenti. Un albero binario è simmetrico se il sottoalbero sinistro della radice è un'immagine *speculare* del sottoalbero destro della radice.

Negli esempi successivi, l'albero binario T_1 non è simmetrico, mentre T_2 lo è.



Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Algoritmi e Strutture Dati – Parte B - 24/07/2020

Esercizio B1, punti ≥ 8

Scrivere un algoritmo che prenda in input tre stringhe X, Y, Z di lunghezza n_x, n_y, n_z e restituisca la lunghezza della più lunga sottosequenza che sia comune alle tre stringhe.

Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Esercizio B2, punti ≥ 10

Siete incaricati del difficile compito di organizzare la didattica del primo semestre 20/21, ma purtroppo c'è una pandemia in atto.

- Avete a disposizione un insieme \mathcal{A} di n aule.
- Ogni aula può essere occupata per 25 slot $\mathcal{S} = \{1, \dots, 25\}$: 5 slot al giorno da due ore (9.00, 11.00, 13.00, 15.00, 17.00), 5 giorni alla settimana.
- Dovete allocare un insieme \mathcal{C} contenente m corsi.
 - Ogni corso $c_i \in \mathcal{C}$ può essere svolto in un sottoinsieme di aule $A_i \subseteq \mathcal{A}$, per via della loro capienza.
 - Ogni corso $c_i \in \mathcal{C}$ può essere svolto in un sottoinsieme di slot $S_i \subseteq \mathcal{S}$, per evitare sovrapposizioni (ad esempio, ASD può essere svolto il mercoledì pomeriggio, ma non la mattina perchè c'è Database).
 - Ogni corso può essere allocato in zero, uno, o due slot anche non consecutivi, ma non più di due. Le altre ore verranno svolte online.

Ogni corso c_i appartiene ad una sola triennale $t[i]$ appartenente all'insieme \mathcal{T} contenente t triennali. Ad ogni triennale possono essere assegnate al massimo k slot alla settimana, per par condicio.

Descrivere un algoritmo che riesca ad allocare il maggior numero di slot nelle aule, rispettando i vincoli.

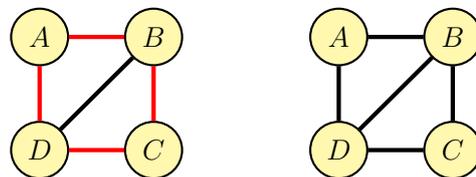
Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Esercizio B3, punti ≥ 12

Scrivere un algoritmo che prenda in input un grafo orientato G e stampi tutti i cicli hamiltoniani presenti in esso. Vi ricordo che un *ciclo hamiltoniano* è un ciclo semplice senza vertici ripetuti che comprende tutti i nodi di un grafo.

Discutere correttezza e complessità computazionale dell'algoritmo proposto.

Nella figura seguente, la parte sinistra mostra un grafo orientato ed un ciclo hamiltoniano evidenziato in rosso. La parte destra mostra un grafo orientato in cui non è presente un ciclo hamiltoniano.



Nota aggiuntiva: Si noti che i cammini A-B-C-D, B-C-D-A, C-D-A-B e D-A-B-C sono lo stesso cammino, stampato da quattro punti di partenza diversi. Se riuscite a stamparne uno solo, avrete punteggio pieno. Se li stampate tutti e quattro, avrete un malus di -10%. Quindi, nel grafo di sinistra l'algoritmo dovrà semplicemente stampare uno dei quattro cammini di cui sopra, nel grafo di destra nessun cammino.