Algoritmi e Strutture Dati (modulo I) - testo prova scritta 25/07/2023
docenti: Luciano Gualà & Andrea Clementi

Cognome: SCURSATONE Nome: CRISTIAN Matr. 0322057

Esercizio 1 [16 punti]
A: notazione asintotica. Dire quali delle seguenti relazioni asintotiche sono vere:

$n + n^2 \log^2 n = o(n^2 \log n)$; $\log^4 n = o(\sqrt[3]{n})$; $n^2 = \Omega(\frac{n^2}{\log^{2001} n})$; $\frac{n\sqrt{n+\log n}}{\sqrt{n^3+3}} = \Theta(\log n)$;
$2^{2n} = \omega(2^{1.9n})$; $2^n = \Theta(2^n + 1.5^n)$; $2^n = o(2^n + n^2)$; $2^n = \Theta(2^{n+8})$;

B: equazioni di ricorrenza. Fornire la soluzione asintotica alle seguenti relazioni di ricorrenza:
$T(n) = T(\frac{99}{100}n) + n$; Soluzione:
$T(n) = T(n-1) + n^3$; Soluzione:

C: algoritmi e complessità. Quale algoritmo useresti e quanto costa se devi:

- In un grafo diretto dire se esiste un nodo t che non può essere raggiunto da almeno un nodo s:
- In un grafo non orientato e pesato, individuare il cammino più corto da s a t che non passa per uno specifico nodo w:
- Costruire un albero AVL contenente n chiavi fornite in input:
- Fondere due heap binari, uno contenente n^2 nodi e l'altro n nodi:

Esercizio 2 [8 punti]
Sia $A[1:n]$ un vettore di n numeri. Un *k-picco* in A è un indice $m \in \{k+1, \ldots, n-k\}$ tale che la sequenza di elementi $A[m-k; m]$ è strettamente crescente, mentre la sequenza di elementi $A[m; m+k]$ è strettamente decrescente.

Si progetti un algoritmo che dato A calcola il più grande valore di k per cui A contiene un k-picco. L'algoritmo deve avere complessità $O(n)$. Si fornisca lo pseudocodice dettagliato.

Esercizio 3 [8 punti]
Nell'ultimo gioco rilasciato dalla *Mintendo*, Super Ciano Bross si trova su un nodo s di un grafo orientato $G = (V, E)$ con n nodi ed m archi, e deve raggiungere il nodo t per vincere il livello. Ogni arco e è associato inizialmente uno *stato* $\sigma(e) \in \{\texttt{on}, \texttt{off}\}$. Super Ciano può attraversare solo gli archi che sono nello stato `on`. C'è inoltre un insieme di nodi $B \subseteq V$ che contengono un bottone speciale. Se Ciano è su un nodo $b \in B$ può decidere di schiacciare il bottone e tutti gli archi invertono il proprio stato, quelli che erano nello stato `on` passano allo stato `off` e quelli che erano nello stato `off` passano nello stato `on`.

Progettate un algoritmo di complessità $O(m+n)$ che calcola, se esiste, una strategia per Super Ciano che lo porta a vincere il livello nel nodo t.

1)

(a)

- $\frac{n + n^2 \lg^2(n)}{n^2 \lg n} = o(n^2 \lg n)$ FALSO

- $\lg^4(n) = o(\sqrt[3]{n})$ VERO

- $n^2 = \Omega\left(\frac{n^2}{\lg^{2001} n}\right)$ VERO

- $\frac{n\sqrt{n + \lg n}}{\sqrt{n+3}} = \Theta(\lg n)$ FALSO

- $2^{2n} = \omega(2^{1.9n})$ VERO

- $2^n = \Theta(2^n + 1{,}5^n)$ FALSO

- $2^n = o(2^n + n^2)$ VERO

- $2^n = \Theta(2^{n+8})$ VERO

(b)

(1) $T(n) = T(n-1) + n^3$

ALBERO UNARIO

$$\left.\begin{array}{c} n \\ | \\ n-1 \\ | \\ \vdots \\ 1 \end{array}\right\} \; n \text{ VOLTE}$$

COSTO SINGOLA ISTRUZIONE $\leq n^3$

$T(n) = O(n \cdot n^3) = O(n^4)$

MA SAPPIAMO CHE LA 1° METÀ OGNI ISTRUZIONE $\geq \frac{n^3}{2}$

QUINDI $T(n) \geq \frac{n}{2} \cdot \frac{n^3}{2} = \frac{n^4}{2} = \Omega(n^4) \Rightarrow T(n) = \Theta(n^4)$

② $T(n) = T\left(\frac{99}{100} n\right) + n =$

$$= T\left(\left(\frac{99}{100}\right)^2\right) + n + \frac{99}{100} n \quad \dots$$

$$= n\left(1 + \frac{99}{100} + \left(\frac{99}{100}\right)^2 + \left(\frac{99}{100}\right)^3 + \dots\right)$$

MA SERIE GEOM. $\sum_{k=0}^{\infty} q^k = \frac{1}{1-q} = \frac{1}{1-\frac{99}{100}} = \frac{1}{\frac{1}{100}} = 100$

QUINDI

$T(n) \cong n \cdot 100$ 100 COSTANTE, QUINDI

$$T(n) = \Theta(n)$$

ⓒ

- DA 1 $V = (e\ell, k)$, CHE SARÀ LA CODA INIZIALE, ESEGUO $n-1$ INSERT(e, k), DALLA CLASSE CODA CON PRIORITÀ

COSTO $\Rightarrow O(n \cdot \lg_d(n)) = O(n \cdot \lg n)$

- VISITA BFS DA SORGENTE s SU UN GRAFO G' CHE È LA COPIA DI G MA SENZA IL NODO w E I SUOI ARCHI (ENTRANTI E USCENTI). DOPO SI VERIFICA SE LA DISTANZA TRA s E t SIA $\leq k$.

COSTO $\Rightarrow$ · RIMOZIONE w E REL. ARCHI $\rightarrow O(1)$

· VISITA DFS $\Rightarrow$ LISTE ADIACENZA $\Rightarrow O(n+m)$

• SE ORDINATA CRESCENTE → RICERCA LINEARE, SCORRIMENTO DA PUNTATORE A PUNTATORE CON K COME CONTEGGIO LIMITE

• SE ORDINATA DECRESCENTE → SCORR. TUTTO A DESTRA (ULTIMO EL.) E POI RIC. LINEARE

COSTO ⇒ $\Theta(n)$

• ALGO → MERGE(C_1, C_2) SI USA PROPRIETÀ IMPL. U HEAP BINOMIALI.

COSTO

$$\log(n+n^2) \cong \log(n^2) = 2\log n = O(\log n)$$

$$T(n) = O(\log n)$$

$\boxed{2}$

K = 2

+ GRA K ?

PER TROVARE IL VALORE K + ALTO, BASTA TROVARE IL PICCO PIÙ ALTO NELL'ARRAY, CIOÈ UN PICCO DOVE NEL SUO INDICE CENTRALE m, GLI ELEMENTI IN $[m-k, m]$ È STRETT. CRESCENTE, E GLI ELE. IN $[m, m+k]$ STRETTAMENTE CRESCENTE, BISOGNA QUINDI TROVARE IL PICCO CHE HA IL VALORE K + GRANDE.

IDEA: SCORRERE L'ARRAY, E TROVO ITERATIVAMENTE
IL PICCO + GRANDE

Esercizio 2 [8 punti]
Sia $A[1:n]$ un vettore di n numeri. Un k-*picco* in A è un indice $m \in \{k+1, \ldots, n-k\}$ tale che la sequenza di elementi $A[m-k; m]$ è strettamente crescente, mentre la sequenza di elementi $A[m; m+k]$ è strettamente decrescente.

Si progetti un algoritmo che dato A calcola il più grande valore di k per cui A contiene un k-picco. L'algoritmo deve avere complessità $O(n)$. Si fornisca lo pseudocodice dettagliato.

Esercizio 3 [8 punti]
Nell'ultimo gioco rilasciato dalla *Mintendo*, Super Ciano Bross si trova su un nodo s di un grafo orientato $G = (V, E)$ con n nodi ed m archi, e deve raggiungere il nodo t per vincere il livello. Ogni arco e è associato inizialmente uno *stato* $\sigma(e) \in \{\texttt{on}, \texttt{off}\}$. Super Ciano può attraversare solo gli archi che sono nello stato `on`. C'è inoltre un insieme di nodi $B \subseteq V$ che contengono un bottone speciale. Se Ciano è su un nodo $b \in B$ può decidere di schiacciare il bottone e tutti gli archi invertono il proprio stato, quelli che erano nello stato `on` passano allo stato `off` e quelli che erano nello stato `off` passano nello stato `on`.

Progettate un algoritmo di complessità $O(m+n)$ che calcola, se esiste, una strategia per Super Ciano che lo porta a vincere il livello nel nodo t.

3 POSSIAMO CONSIDERARE LO "SWITCH" COME UN CAMBIO DI LIVELLO.

<u>IDEA</u>: CREIAMO, A PARTIRE DA G, UN GRAFO $G' = (V', E')$.

DOVE:

$$V' = \underbrace{V \cup V^{CP}}_{} \hookrightarrow = \{u' : u \in V\} \cup t''$$

$$E' = \{ e \in E : \forall u,v \in V : (u,v) \in E' \iff w(e) = ON \;\wedge$$

$$\forall u',v' \in V^{CP} : (u',v') \in E' \iff w(e) = OFF.$$

$$\cup \{(c,c') \in E \iff c \in B\} \cup (t,t'') \cup (t',t'')$$

t'' NUOVO NODO FINALE

QUINDI UN GRAFO COMPOSTO DA 2 COPIE DI G DOVE 1 CONTIENE SOLO ARCHI CON STATO ON E L'ALTRA CON ARCHI CON STATO OFF.

E $\forall$ NODO $\in B$, $\exists$ 1 ARCO BIDIREZIONALE CHE VA DA e ALLA SUA COPIA.