Algoritmi e Strutture Dati (modulo I) - testo prova scritta 27/09/2023

docenti: Luciano Gualà & Andrea Clementi

Cognome:........................... Nome:............................... Matr.:...............

Esercizio 1 [16 punti]

A: notazione asintotica. Dire quali delle seguenti relazioni asintotiche sono vere:

$n + n\sqrt{n}\log^2 n = \Theta(n^{1.5})$; $\log^3 n = \Theta(\log n)$; $n + \sqrt{n} = \Theta(n - \sqrt{n})$; $\frac{n^{1.5}\sqrt{n+\log n}}{\sqrt{n^3+3}} = \Theta(\sqrt{n})$;

$(\frac{5}{3})^n = o(2^n)$; $2^n = o(2^n\sqrt{\log n})$; $2^n = \Theta(2^{2n})$; $2^{n+2} = \Theta(2^n)$;

B: equazioni di ricorrenza. Fornire la soluzione asintotica alle seguenti relazioni di ricorrenza:

$T(n) = 7\,T(\frac{n}{8}) + n$; Soluzione:

$T(n) = T(n-1) + n^2$; Soluzione:

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

- Costruire un heap binario contenente n chiavi prese in input:
- In un grafo orientato, capire se c'è un cammino da s a t di al più k archi che evita uno nodo specifico w:
- Trovare il k-esimo minimo in una lista ordinata di n elementi (implementata con record e puntatori):
- Fondere due heap binomiali contenenti rispettivamente n e n^2 nodi:

Esercizio 2 [8 punti]

Un labirinto è modellato come un grafo non diretto $G = (V, E)$. Voi siete nel nodo s e l'uscita si trova nel nodo t. Potete percorrere gli archi, spendendo un minuto per ogni arco. Nel labirinto inoltre c'è un nodo speciale p che è un teletrasporto, e un insieme di nodi $U \subseteq V$ che sono uscite del teletrasporto. Se siete su p potete teletrasportarvi in un qualsiasi nodo $q \in U$ a vostra scelta. Il tempo del teletrasporto è di 3 minuti.

Progettate un algoritmo efficiente che calcoli la strategia più veloce, se esiste, per uscire dal labirinto.

Esercizio 3 [8 punti]

Sia $A[1:n]$ un vettore di n bit, dove quindi $A[i] \in \{0,1\}$ per ogni i. Si progetti una struttura dati che prende in input il vettore A e sia in grado poi di rispondere a query del seguente tipo:

- `BlockSize(i)`: dato un indice $i \in \{1, 2, \ldots, n\}$ restituisce la lunghezza del più grande blocco di zero contigui che contiene l'indice i. Se $A[i] = 1$, allora la risposta alla query è 0.

La struttura dati deve poter essere costruita in tempo $O(n)$ e l'algoritmo di query deve richiedere tempo constante. Si forniscano i due pseudocodici dettagliati dell'algoritmo che dato A costruisce la struttura dati, e dell'algoritmo di query.

1 a

A: notazione asintotica. Dire quali delle seguenti relazioni asintotiche sono vere:
$n + n\sqrt{n}\log^2 n = \Theta(n^{1.5})$; F $\log^3 n = \Theta(\log n)$; F $n+\sqrt{n} = \Theta(n-\sqrt{n})$; V $\frac{n^{1.5}\sqrt{n+\log n}}{\sqrt{n^3+3}} = \Theta(\sqrt{n})$; V
$(\frac{5}{3})^n = o(2^n)$; ✓ $2^n = o(2^n\sqrt{\log n})$; ✓ $2^n = \Theta(2^{2n})$; F $2^{n+2} = \Theta(2^n)$; ✓

$$\frac{n + n\sqrt{n}\log^2 n}{n\sqrt{n}} = \frac{1}{\sqrt{n}} + \log^2 n \to 0 + \infty = +\infty$$

$$n + n\sqrt{n}\log^2 n = \omega(n^{1.5})$$

$$\frac{n\sqrt{n}\sqrt{n+\log n}}{\sqrt{n^3+\cancel{3}}\cdot\sqrt{n}} = \cancel{\frac{n\sqrt{n}}{n\sqrt{n}}}\,\cancel{\frac{\sqrt{n}}{\sqrt{n}}} = 1$$

$$\frac{n+\sqrt{n}}{n-\sqrt{n}} = \frac{\cancel{\sqrt{n}}(\sqrt{n}+1)}{\cancel{\sqrt{n}}(\sqrt{n}-1)} = \frac{\sqrt{n}+1^{\,-2\,+2}}{\sqrt{n}-1} = = 1 + \frac{2}{\sqrt{n}-1} = = 1$$

FORSE × GERARCHIA INFINITI?

$$\left(\frac{5}{3}\right)^n \cdot \frac{1}{2^n} = \frac{5^n}{3^n \cdot 2^n} = 0 \quad ?$$

$6 > 5$?

FORSE

b

B: equazioni di ricorrenza. Fornire la soluzione asintotica alle seguenti relazioni di ricorrenza:
$T(n) = 7\,T(\frac{n}{8}) + n$; Soluzione:
$T(n) = T(n-1) + n^2$; Soluzione:

$$T(n) = T(n-1) + n^2$$

n ISTRUZIONI, OGNUNO COSTA $\leq n^2$

$$T(n) = O(n \cdot n^2) = O(n^3) \to T(n) = \Theta(n^3).$$

$$T(n) = 7T\left(\frac{n}{8}\right) + n$$

$$n^{\log_8 7} \quad \text{VS} \quad n$$

$$n = \Omega\left(n^{\log_8 7 + (1 - \log_8 7)}\right)$$

$$7\,\frac{n}{8} \le cn \qquad c = \frac{7}{8}$$

CASO 3

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

C

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

- Costruire un heap binario contenente n chiavi prese in input: n INSERT $T(n) = O(n \cdot \log n)$
- In un grafo orientato, capire se c'è un cammino da s a t di al più k archi che evita uno nodo specifico w: VISITA DFS MA TOGLIENDO. IL NODO w IN G E I SUOI ARCHI INCIDENTI ↳ COSTO $= O(n + m)$.
- Trovare il k-esimo minimo in una lista ordinata di n elementi (implementata con record e puntatori): VISITA LINEARE ↳ COSTO $= O(n)$
- Fondere due heap binomiali contenenti rispettivamente n e n^2 nodi:

↳ $m = n + n^2 = O(n^2)$

COSTO FUSIONE $\to O(\log m) = O(\log n^2) =$

$= O(2 \log n)$.

2)

Esercizio 2 [8 punti]
Un labirinto è modellato come un grafo non diretto $G = (V, E)$. Voi siete nel nodo s e l'uscita si trova nel nodo t. Potete percorrere gli archi, spendendo un minuto per ogni arco. Nel labirinto inoltre c'è un nodo speciale p che è un teletrasporto, e un insieme di nodi $U \subseteq V$ che sono uscite del teletrasporto. Se siete su p potete teletrasportarvi in un qualsiasi nodo $q \in U$ a vostra scelta. Il tempo del teletrasporto è di 3 minuti.

Progettate un algoritmo efficiente che calcoli la strategia più veloce, se esiste, per uscire dal labirinto.

3

Esercizio 3 [8 punti]
Sia $A[1:n]$ un vettore di n bit, dove quindi $A[i] \in \{0,1\}$ per ogni i. Si progetti una struttura dati che prende in input il vettore A e sia in grado poi di rispondere a query del seguente tipo:

- `BlockSize(i)`: dato un indice $i \in \{1, 2, \ldots, n\}$ restituisce la lunghezza del più grande blocco di zero contigui che contiene l'indice i. Se $A[i] = 1$, allora la risposta alla query è 0.

La struttura dati deve poter essere costruita in tempo $O(n)$ e l'algoritmo di query deve richiedere tempo constante. Si forniscano i due pseudocodici dettagliati dell'algoritmo che dato A costruisce la struttura dati, e dell'algoritmo di query.

```
ORACOLO (A):

  Y UN ARRAY AUSILIARIO DI LUNGHEZZA DI A.
  C=0
  FOR i=1 TO n DO

      IF A[i] = 1
          C = 0
          Y[i] = C
      ELIF A[i] = 0
          C++
          Y[i]=C

  FOR i=n-1 DOWN TO 1 DO

      IF A[i] = A[i+1] = 0

          Y[i] = Y[i+1]
  RETURN Y

QUERY (i)

  RETURN Y[i].
```