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:

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

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

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

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

$T(n) = T(n-1) + n^2$; Soluzione: $\Theta(n^3)$

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

- Costruire un heap binario contenente n chiavi prese in input: heapify $O(n)$
- 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:
- Trovare il k-esimo minimo in una lista ordinata di n elementi (implementata con record e puntatori): $O(n)$ ricerca lineare
- Fondere due heap binomiali contenenti rispettivamente n e n^2 nodi: $O(\log n^2)$ merge( ) ristruttura $O(\log(n))$

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.

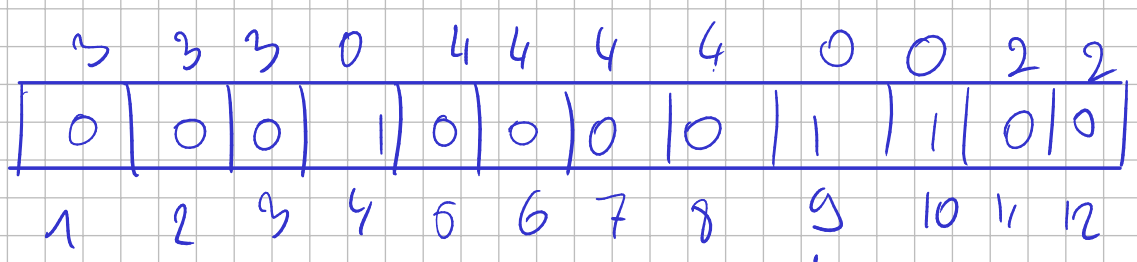
Progettate un algoritmo efficiente che calcoli la strategia più veloce, se esiste, per uscire dal labirinto.

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.



creaOracolo(A)

Sia Y un array di dim n

```
c = 0
for i = 1 to n do
    if A[i] == 0
        c++
        Y[i] = c
    else if A[i] == 1
        c = 0
        Y[i] = c
```

```
for c = n-1 to 1 do
    if A[i] == 1
        Y[i] = 0
    if A[i] == 0 AND
        A[i+1] == 1
            continue
    else Y[i] = Y[i+1]
```