

## Algoritmi e Strutture Dati – Parte A - 13/7/22

**Esercizio -1** Iscriverti allo scritto entro la scadenza. In caso di inadempienza, -1 al voto finale.

**Esercizio 0** Scrivere correttamente nome, cognome, numero di matricola, riga e colonna su tutti i fogli consegnati. Consegnare foglio A4 e foglio protocollo di bella. In caso di inadempienza, -1 al voto finale.

### Esercizio A1 – Complessità – Punti $\geq 8$

Si consideri la seguente equazione di ricorrenza:

$$T(n) = \begin{cases} T(\lfloor 2n/9 \rfloor) + T(\lfloor 5n/18 \rfloor) + n & n > 1 \\ 1 & n \leq 1 \end{cases}$$

Individuare limiti inferiori e superiori tramite il metodo di sostituzione.

### Esercizio A2 – Lago ricorsivo – Punti $\geq 10$

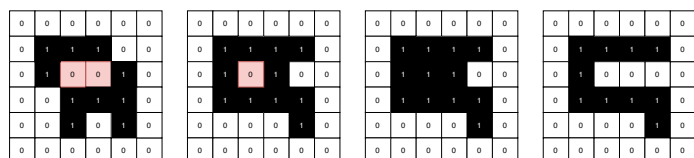
Sia  $M$  una matrice booleana  $n \times n$  che rappresenta una mappa, dove le caselle 1 sono terreno e le caselle 0 sono acqua. La mappa rappresenta un tratto di mare che contiene una e una sola isola; per comodità, potete assumere che le caselle lungo i bordi abbiano valore 0 (siano acqua). Un'isola è un insieme contiguo (in verticale o orizzontale) di caselle 1, eventualmente contenenti caselle 0 al proprio interno. Qualcuno afferma che dentro l'isola ci sia un lago.<sup>1</sup> Un lago è un'insieme contiguo (in verticale o orizzontale) di caselle 0, non connesso al mare orizzontalmente o verticalmente. Il vostro compito è scrivere un algoritmo

**boolean** containsLake(**int**[][]  $M$ , **int**  $n$ )

che prende in input la matrice  $M$  e la dimensione  $n$ , e restituisce **true** se l'isola contiene uno o più laghi.

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

I quattro esempi seguenti rappresentano quattro test case in cui l'algoritmo deve restituire **true**, **true**, **false**, **false**. Le isole sono evidenziate in rosa.



### Esercizio A3 – H-index – Punti $\geq 12$

L'indice H, o indice di Hirsch (h-index) è una misura di impatto della produzione scientifica dei ricercatori, basato sul numero di pubblicazioni e sul numero di citazioni.<sup>2</sup> L'h-index di un ricercatore è pari  $h$  se  $h$  articoli tra quelli che ha pubblicato sono stati citati almeno  $h$  volte ciascuno e non è vero che  $h + 1$  articoli fra quelli che ha pubblicato sono stati citati almeno  $h + 1$  volte.

Scrivere un algoritmo

**int** h-index(**int**[]  $A$ , **int**  $n$ )

che prende in input un vettore contenente  $n$  interi positivi ordinati in maniera non crescente, contenente il numero di citazioni ricevute dagli  $n$  articoli pubblicati da un ricercatore, e restituisce il suo h-index.

Discutere informalmente la correttezza della soluzione proposta e calcolare la complessità computazionale. *Soluzioni in tempo lineare o superiore non verranno accettate.*

Per esempio,

- se  $A = [16, 12, 7, 5, 4, 3, 2, 1]$ , l'h-index è pari a 4, in quanto i primi 4 articoli hanno ricevuto almeno 4 citazioni, ma i primi 5 articoli non hanno ricevuto 5 citazioni;
- se  $A = [54, 17, 14, 10, 9, 6, 6, 5, 4, 3]$ , l'h-index è pari a 6, in quanto i primi 6 articoli hanno ricevuto almeno 6 citazioni. Il settimo ha ricevuto 6 citazioni, quindi non viene contato.

<sup>1</sup>Un'isola ricorsiva è un'isola che si trova all'interno di un lago che si trova all'interno di un'isola. Un lago ricorsivo è un lago che si trova all'interno di un'isola che si trova all'interno di un lago. Esistono numerosi esempi di questo tipo, e la ricorsione può andare anche oltre il primo livello: l'isola Manar è un'isola che si trova in lago che si trova in isola che si trova in un lago che si trova in un'isola che si trova in un lago in Canada. Per ulteriori esempi di questo tipo, fate riferimento alla pagina [https://en.wikipedia.org/wiki/Recursive\\_islands\\_and\\_lakes](https://en.wikipedia.org/wiki/Recursive_islands_and_lakes).

<sup>2</sup>Serve a ben poco, se non a creare sciocchi battibecchi fra ricercatori in cui uno può punzecchiare l'altro dicendogli "Il mio h-index è più grosso del tuo!"

**Algoritmi e Strutture Dati – Parte B - 13/7/22**

**Esercizio -1** Iscriverti allo scritto entro la scadenza. In caso di inadempienza, -1 al voto finale.

**Esercizio 0** Scrivere correttamente nome, cognome, numero di matricola, riga e colonna su tutti i fogli consegnati. Consegnare foglio A4 e foglio protocollo di bella. In caso di inadempienza, -1 al voto finale.

**Esercizio B1 – Prodotto minimo – Punti  $\geq 8$** 

Scrivere un algoritmo

```
int minproduct(int[] A, int n)
```

che prende in input un vettore  $A$  contenente  $n$  interi e restituisce il minimo prodotto che può essere ottenuto moltiplicando un sottoinsieme non vuoto di valori di  $A$  fra di loro.

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

Per esempio,

- per  $A = [-1, -1, 2, -2, 4, 3]$ , l'output è  $-1 \cdot -1 \cdot 2 \cdot 2 \cdot -2 \cdot 4 \cdot 3 = -48$ ;
- per  $A = [-1, 0]$  l'output è  $-1$  (valore singolo)
- per  $A = [0, 1]$  l'output è  $0$  (valore singolo)
- per  $A = [1, 2, 4]$ , l'output è  $1$  (valore singolo)

**Esercizio B2 – Stampa con spazi – Punti  $\geq 10$** 

Scrivere un algoritmo

```
printWithSpaces(ITEM[] A, int n)
```

che prende in input una stringa contenente  $n$  caratteri, e produce in output tutti i modi in cui è possibile stampare la stringa inserendo zero o più caratteri spazio al suo interno, evitando di inserire due spazi consecutivi.

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

Per esempio, se  $A = \text{"CIAO"}$ , l'algoritmo deve stampare

```
CIAO
CIA O
CI AO
CI A O
C IAO
C IA O
C I AO
C I A O
```

non necessariamente in questo ordine.

**Esercizio B3 – Sequenze crescenti – Punti  $\geq 12$** 

Scrivere un algoritmo

```
int countIncreasing(int[] A, int n, int k)
```

che prende in input un vettore  $A$  contenente  $n$  interi distinti e un parametro  $k$ , e restituisce il numero di sottosequenze crescenti contenute nel vettore  $A$  di dimensione  $k$ .

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

Per esempio, se  $A = [-3, 7, 5, 6, 9]$  ( $n = 5$ ) e  $k = 3$ , le sottosequenze crescenti sono:

```
[-3, 7, 9], [-3, 5, 6], [-3, 5, 9], [-3, 6, 9], [5, 6, 9]
```

e quindi l'algoritmo deve restituire 5.