

# Esame di algoritmi e strutture dati

7 luglio 2020

Tempo a disposizione: 2 ore

## Esercizio 1

(7 punti)

1. Progettare un algoritmo (pseudocodice):

$alternante(Lista[Interi] L) \rightarrow Booleano$

che, presa in input una lista  $L$  contenente valori interi, restituisca  $true$  se e solo ciascun valore pari in essa contenuto è seguito da uno dispari e ciascun valore dispari è seguito da uno pari. Se  $L$  è la lista vuota, la funzione deve restituire il valore  $true$ .

Esempio:

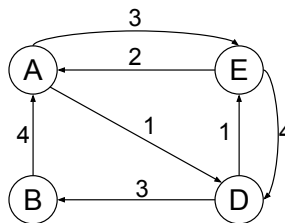
- per  $L_1 = 4, 1, 0, 3, 8$ ,  $alternante(L_1) = true$
- per  $L_1 = 3, 2, 11, 4, 1$ ,  $alternante(L_1) = true$
- per  $L_1 = 0, 0, 4, 3, 8$ ,  $alternante(L_1) = false$

2. Determinare il costo temporale dell'algoritmo.

## Esercizio 2

(7 punti)

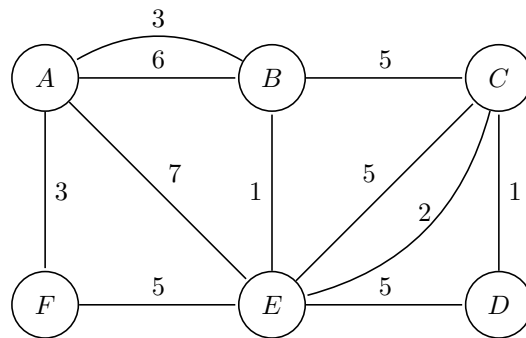
- Descrivere in una frase a che scopo viene utilizzato l'algoritmo di Bellman e Ford.
- Mostrare lo pseudocodice dell'algoritmo e la sua complessità.
- Illustrare l'esecuzione dell'algoritmo sul seguente input, prendendo come sorgente il nodo  $A$ :



### Esercizio 3

(6 punti)

1. Fornire la definizione di *minimo albero ricoprente*.
2. Fornire la definizione di *albero dei cammini minimi*.
3. Dato il grafo sotto riportato, mostrarne un minimo albero ricoprente ed un albero dei cammini minimi.



### Esercizio 4

(6 punti)

1. Fornire la definizione di *Albero Binario di Ricerca (BST)*.
2. Mostrare lo pseudocodice dell'algoritmo di ricerca in un BST.
3. Dimostrare formalmente il costo temporale dell'algoritmo nel caso peggiore.
4. Mostrare un esempio di "caso peggiore" e discutere se esistano strutture dati che permettano di migliorare le prestazioni della ricerca in un BST.

### Esercizio 5

(7 punti)

Si consideri il seguente insieme di studenti, ciascuno associato al proprio numero di matricola:

Nome	Matricola
Marco	11765
Paola	21543
Andrea	14278
Giovanni	13369
Francesca	21444
Riccardo	11899
Paola	31363
Giuseppe	21722
Martina	16656
Sara	13479

Considerando la matricola come chiave, si supponga di voler organizzare questo insieme (dizionario) in una tabella Hash di dimensione  $m = 2$ , gestita con liste di collisione.

1. Definire un'opportuna funzione Hash  $h(k)$ .
2. Indicare, motivando la risposta, se la funzione scelta sia perfetta o meno.
3. Mostrare lo stato della tabella dopo l'inserimento, nell'ordine riportato, dei seguenti elementi:

$\langle Sara, 13479 \rangle \langle Paola, 21543 \rangle \langle Marco, 11765 \rangle$

4. Illustrare, argomentando opportunamente, quale sia il costo temporale dell'inserimento di un elemento, nel caso peggiore.