

Nome e Cognome .....

Matricola.....

## Esame Scritto di Algoritmi 2 del 15/02/2021

Non è consentito l'uso di libri, appunti e qualsiasi altro materiale. L'esame dura 2 ore.

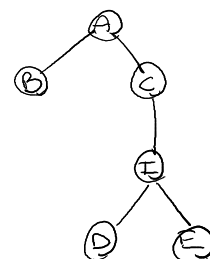
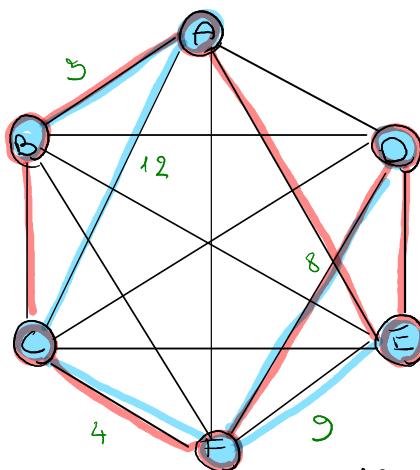
È possibile consegnare o ritirarsi prima della fine dell'esame (per ritirarsi, scrivere RITIRATO in cima a questo foglio e a quello protocollo). Il punteggio totale è 32 (30 e lode).

### ESERCIZIO 1. (Punteggio 8 punti)

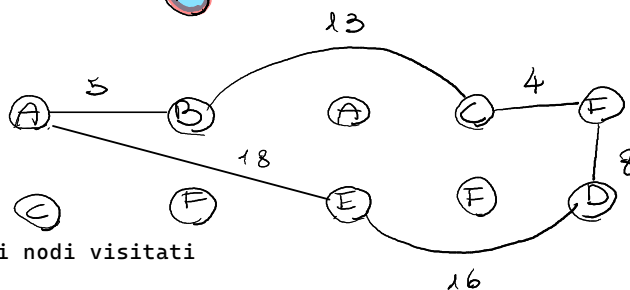
Utilizzando l'algoritmo approssimato visto a lezione si trovi, per il seguente grafo rappresentato con matrice di adiacenza, un ciclo Hamiltoniano di peso al più 2 volte il peso del cammino Hamiltoniano di peso minimo.

Ovunque sia possibile effettuare una scelta arbitraria, si scelgano i vertici in ordine alfabetico. Si descrivano i passaggi fatti.

	A	B	C	D	E	F
A	0	5	12	17	18	11
B	5	0	13	12	15	12
C	12	13	0	11	13	4
D	17	12	11	0	16	8
E	18	15	13	16	0	9
F	11	12	4	8	9	0



ABACFDFEFC



1. Trovo MAR (con Prim o Kruskal)
2. Visito MAR con DFS e costruisco grafo dei nodi visitati
3. Rimuovo vertici duplicati

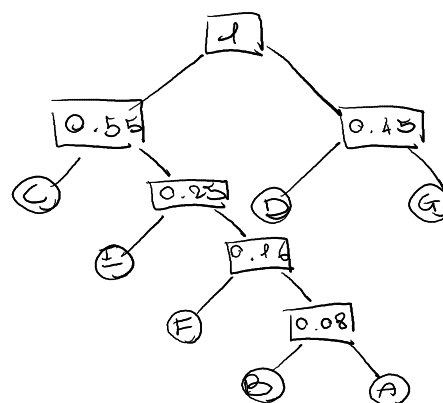
### ESERCIZIO 2. (Punteggio 7 punti)

1. Dato l'alfabeto composto dai caratteri **a, b, c, d, e, f, g** e la seguente tabella delle frequenze, si calcoli una codifica binaria a lunghezza variabile dell'alfabeto secondo l'algoritmo di Huffman. (Si mostri come la struttura mantenuta dall'algoritmo cambia ad ogni iterazione)

Carattere	a	b	c	d	e	f	g
Frequenza	0.06	0.02	0.30	0.20	0.09	0.08	0.25

2. Quale tecnica algoritmica adotta l'algoritmo di Huffman?  
Greedy, appetibilità modificabili

PQ : 0.02 0.06 0.08 0.09 0.20 0.25 0.30  
 PQ : 0.02 0.08 0.09 0.26 0.26 0.30  
 PQ : 0.09 0.16 0.26 0.26 0.30  
 PQ : 0.20 0.25 0.25 0.30  
 PQ : 0.25 0.30 0.45  
 PQ : 0.45 0.55



### ESERCIZIO 3. (Punteggio 6 punti)

- Utilizzando l'algoritmo visto a lezione, trovare la più lunga sottosequenza comune (LCS) tra le stringhe "FLILFAI" e "LIFIAI".
- Quale tecnica usa questo algoritmo? (si indichino anche eventuali sottocategorie)

Programmazione dinamica

matrice LCS

		F	L	I	L	F	A	I
L		←	↖	←	↖	←	←	←
I		↑	↑	↖	←	←	←	←
F		↖	↑	↑	↑	↖	←	←
I		↑	↑	↖	↑	↑	↑	↖
A		↑	↑	↑	↑	↑	↖	↑
I		↑	↑	↖	↑	↑	↑	↖

matrice L

		F	L	I	L	F	A	I
		0	0	0	0	0	0	0
L		0	0	1	1	1	1	1
I		0	0	1	2	2	2	2
F		0	1	1	2	2	3	3
I		0	1	1	2	2	3	4
A		0	1	1	2	2	3	4
I		0	1	1	2	2	3	4

LCS("FLILFAI", "LIFIAI") = LIFAI

### ESERCIZIO 4. (Punteggio 7 punti)

- Scrivere (in pseudocodice) un algoritmo che, dato un grafo non pesato e orientato  $G$ , rappresentato con liste di adiacenza ed eventualmente non connesso, restituisca un insieme ordinato di archi  $C$ , appartenenti al grafo, che nell'ordine formano un ciclo all'interno del grafo (se il grafo non contiene cicli, viene restituito un insieme vuoto).  
NOTAZIONE: Per inserire (eventualmente rimuovere) elementi da  $C$ , si usino i classici operatori per le liste (es.  $\text{addFirst}(C, \langle u, v \rangle)$ ,  $\text{addLast}(C, \langle u, v \rangle)$ , ...).

### ESERCIZIO 5. (Punteggio 4 punti)

Si dica se le seguenti affermazioni sono vere (V) o false (F). (motivare tutte le risposte)

- La programmazione dinamica adotta tecniche Top-down.
  - Un algoritmo non deterministico è un algoritmo NP-completo.
  - Con l'algoritmo di Bellman-Ford è possibile rilevare cicli negativi.
- Falso. Adotta tecniche bottom-up, in quanto il problema è diviso in ulteriori sottoproblemi.
  - Falso. Un problema può essere NP-completo, non un algoritmo.
  - Vero. Basta controllare se dopo l'esecuzione di  $n-1$  cicli esterni la distanza di qualche vertice diminuisce.

#### Esercizio 4

```

C ← lista vuota
for ogni nodo in G
    if color[u] = white and VISITA-RIC-CICLO(G,u,C)
        return C
removeAll(C)
return C

```

```

color[u] ← gray
addFirst(C,v)
for ogni v adiacente a u
    if color[v] = white
        padre[v] ← u
        if VISITA-RIC-CICLO(G,v,u) return true
    else if color[v] = gray return true
color[u] ← black
return false

```