ANNO ACCADEMICO 2019/2020 - Algoritmi e Strutture Dati - 28 luglio 2020
TURNO A

n. quesito	1	2	3	4	voto
punti	8	8	8	9	33

quesito 1

Calcolare la complessità del seguente comando. Indicare per esteso le relazioni di ricorrenza e, per ogni comando ripetitivo, il numero di iterazioni e la complessità della singola iterazione.

```
for (int i=1; i <=f(g(n)); i++) a++;


int f(int x) {
     if (x==1) return 2;
         else return 2*x+ f(x-1);
}
int g(int x) {
     if (x<=1) return 1;
     int a=f(x); int b=0;
     for (int i=1; i <=a; i++) b++;
     return x+2*g(x/2);
}
```

Funzione f

$T_f(0)= d$
$T_f(n)= c + T_f(n-1)$ $T_f(n)$ è $O(n)$

$R_f(0)= 1$
$R_f(n)= n+ R_f(n-1)$ $R_f(n)$ è $O(n^2)$

Funzione g

$T_g(0,1)=d$
$T_g(n)= cn^2+ T_g(n/2)$ $T_g(n)$ è $O(n^2)$

$R_g(0)=d$
$R_g(n)= n + 2 R_g(n/2)$ $R_g(n)$ è $O(nlogn)$

Calcolo del comando:
numero iterazioni: $R_f(R_g(n)) = R_f(nlogn) = O(n^2 (logn)^2)$
complessità della singola iterazione: $T_g(n) + T_f(nlogn) = O(n^2) + O(nlogn)= O(n^2)$
complessità del for: $O(n^4(logn)^2)$

quesito 2

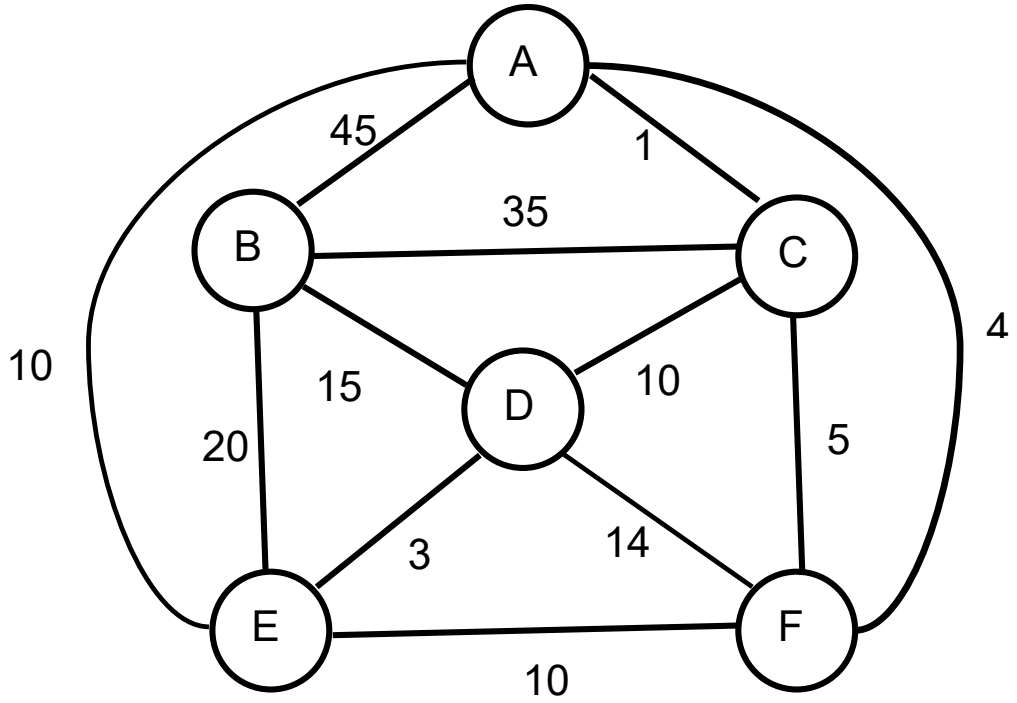
Scrivere una funzione **`void sposta(Node* t, int x)`**, che, dato un albero generico **`t`** con etichette intere tutte diverse memorizzato figlio/fratello e una etichetta **x**, sposta i figli del nodo con etichetta **x** come primo gruppo di figli del fratello successive di **x**. La funzione non fa niente se **x** non compare nell'albero o non ha figli o non ha fratelli successivi.

```
void sposta(Node* t, int x,){
if (!t) return;
if (t->label==x) {
     if (!t->left || !t->right)
       return;
     Node* a=t->left;
     t->left=0;
     Node * b= t->right->left;
     t->right->left=a;
     aggiungi (a,b);
     return;
  }
sposta(t->left, x);
sposta(t->right, x);
}

void aggiungi(Node* t, Node* f)  {
if (!t->right) t->right =f;
aggiungi(t->right);
}
```

quesito 3

a) 1 A cosa serve l'algoritmo di Kruskal?
b) 3 Qual è la sua complessità e come si calcola?
c) 3 Applicarlo al grafo in figura. Mostrare tutti i passaggi (scelta dell'arco e conseguenti insiemi connessi) in una tabella fatta come nel disegno.
d) 1 Indicare un albero di copertura del grafo non minimo.
e) - Tutti gli alberi di copertura di un grafo hanno lo stesso numero di nodi?



Una possibile soluzione

Arco	Componenti connesse
	{A}{B}{C}{D}{E}{F}
(A,C) 1	{A,C}{B}{D}{E}{F}
(E,D) 3	{A,C}{B}{D,E}{F}
(A,F) 4	{A,C,F}{B}{D,E}
(D,C) 10	{A,C,F,D,E}{B}
(B,D) 15	{A,C,F,D,E,B}

quesito 4

a) 3 Dare la definizione di complessità computazionale. Confrontare le funzioni seguenti dal punto di vista della complessità: dire quando una è O dell'altra indicando una coppia (n0,c) e, in caso contrario, spiegare perchè non lo è.

$$F(n) = \begin{cases} 59\ n^3 + n & n \text{ pari} \\ 10\ n^2 & n \text{ dispari} \end{cases}$$

$$G(n) = \begin{cases} 3\ n^4 & n <= 200 \\ 2\ n^3 & \text{altrimenti} \end{cases}$$

Fè O(G): considerare la coppia[n0=201 c=30]
G non è O(F): perché esistono infiniti numero dispari

b) **3** Descrivere il metodo di ricerca hash (accesso diretto, collisioni, agglomerati, metodi di scansione, metodo di concatenazione). Da cosa dipende l'efficienza del metodo?

c) **3** E' possibile sapere a tempo di compilazione di un programma C++ da quale gestore sarà gestita una eventuale eccezione? Se la risposta è affermativa, spiegare perchè, se è negative, fare un esempio. **NO**