

Le domande non sono identiche a quelle del pretest, ma le ho riscritte come me le ricordavo, in particolare per quanto riguarda le opzioni proposte.

Pretest 1° appello estivo 2022
(con risposte in fondo)

1) Complessità per trovare minimo, massimo e valore mediano in un array disordinato

- $O(n)$
- $O(n \log n)$
- $O(n^2)$
- $O(\log n)$

2) Quali tra questi è un algoritmo goloso?

- Dijkstra
- PLSC
- Fattorizzazione
- selectionsort

3) Esercizio in cui trovare cammino minimo tra due nodi in un dato grafo

4) Quale è l'output del seguente programma?

```
#include <iostream>
using namespace std;

template<int n>
class Classe {
    static int istanze;
    int valore;
public:
    Classe() {valore=n; istanze++;}
    void stampa() {cout<<istanze<<' '<<valore<<' ';<<endl;}
};

template<int n>
int Classe<n>::istanze = 0;

int main() {
    Classe<9> uno;
    uno.stampa();

    Classe<7> due;
    due.stampa();

    Classe<9> tre;
    tre.stampa();
    return 0;
}
```

- 1 9 1 7 2 9
- 1 9 2 7 3 9
- 1 9 2 7 2 9
- 1 9 1 7 1 9

5) Quale è l'output del seguente programma?

```
#include <iostream>
using namespace std;

void g(int x){
    try {
        if (x==0) throw 'a';
    } catch (int) {
        cout << "catch g ";
    }
    cout << "fine g ";
}

void f(int x){
    try {
        g(x);
    } catch (char) {
        cout << "catch f ";
    }
    cout << "fine f ";
}

int main() {
    f(0);
    cout << "fine main ";
}
```

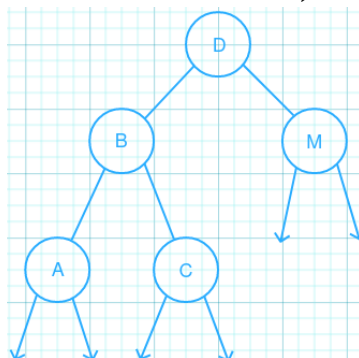
- fine main
- catch f fine f fine main
- catch g fine f fine main
- errore

6) Trovare codifica con algoritmo di Huffman date le frequenze {A:44, B:53, C:12, D:10, E:15, F:3, G:55}

7) Dire il risultato del primo passaggio di radix sort applicato a {BEC, EBA, CEB, BBA, ACE, ABC} con alfabeto {A, B, C, D, E}

8) Trovare la visita (non ricordo in che order) di un albero binario

9) Considerato il seguente albero binario di ricerca (era uno simile) con A valore più piccolo e Z valore più grande (cioè sono in ordine alfabetico crescente), dove viene inserito il nodo X?



- si crea un figlio destro di C
- si crea un figlio sinistro di M
- si crea un figlio destro di M
- si crea un figlio destro di A

10) Definizione albero minimo di copertura

Risposte:

- 1) **nlogn** perché è necessario ordinarlo (nel pretest era stata segnato $O(n)$ come risposta corretta, perché il professore pensava di aver scritto valore medio, e dopo che è stato fatto presente da uno studente sono state considerate corrette entrambe alle verbalizzazioni successive)
- 2) **Dijkstra**
- 3) Con Dijkstra, ma anche ad occhio si vedeva un cammino più breve di quelli tra le opzioni proposte, e quindi **nessuna delle altre**
- 4) **1 9 1 7 2 9** (classi con parametri diversi si comportano come classi diverse)
- 5) **catch f fine f fine main** (viene lanciato un char in g che non viene catturato in g perché il catch è su int, ma viene catturato in f perché il catch è su char)
- 6) Applicando l'algoritmo si ottiene **A:01 B:10 C:0001 D:00001 E:001 F:00000 G:11** (o altri equivalenti). Nel pretest si poteva osservare che in tutte le opzioni alcune lettere con frequenza alta avevano codifiche più lunghe di alcune con frequenza bassa, quindi si potevano tutte escludere e la risposta era **nessuna delle altre**
- 7) **EBA, BBA, CEB, BEC, ABC, ACE**
- 8) Si poteva andare per esclusione guardando già solo i primi nodi
- 9) **Si crea un figlio destro di M** (si scende per l'albero andando a destra se il valore è maggiore dell'etichetta del nodo considerato, altrimenti a sinistra, quando si raggiunge una foglia lo si inserisce come suo figlio)
- 10) **Insieme di componenti connesse massimali tali che la somma dei pesi degli archi è minima**

Pretest 2° appello estivo 2022
(con risposte in fondo)

1) Quale è l'output del seguente programma?

```
#include <iostream>
using namespace std;

template<int n>
class Classe {
    static int istanze;
    int valore;
public:
    Classe() {valore=n; istanze++;}
    void stampa() {cout<<istanze<<' ' <<valore<<' ';}
};

template<int n>
int Classe<n>::istanze = 0;

int main() {
    Classe<9> uno;
    uno.stampa();

    Classe<7> due;
    due.stampa();

    Classe<9> tre;
    tre.stampa();
    return 0;
}
```

- 1 9 1 7 2 9
- 1 9 2 7 3 9
- 1 9 2 7 2 9
- 1 9 1 7 1 9

2) Quale è l'output del seguente programma?

```
#include <iostream>
using namespace std;

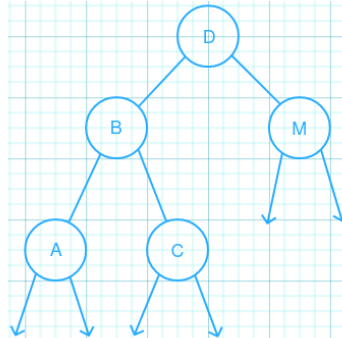
void g(int x){
    try {
        if (x==0) throw 'a';
    } catch (int) {
        cout << "catch g ";
    }
    cout << "fine g ";
}

void f(int x){
    try {
        g(x);
    } catch (int) {
        cout << "catch f ";
    }
    cout << "fine f ";
}

int main() {
    f(0);
    cout << "fine main ";
}
```

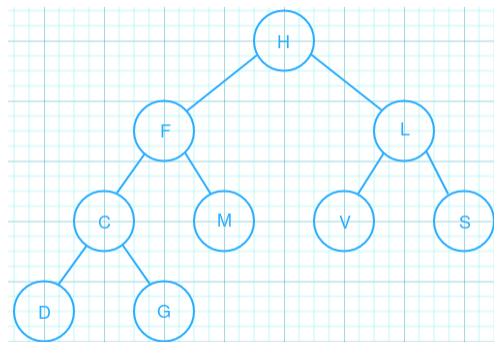
- catch g fine g fine main
- fine g fine f fine main
- catch f fine main
- errore

3) Considerato il seguente albero binario di ricerca (era uno simile) con A valore più piccolo e Z valore più grande (cioé sono in ordine alfabetico crescente), dove viene inserito il nodo X?



- si crea un figlio destro di C
- si crea un figlio sinistro di M
- si crea un figlio destro di M
- si crea un figlio destro di A

4) Visita preorder di un albero binario di ricerca



5) Dire il risultato del primo passaggio di radix sort applicato a {BEC, EBA, CEB, BBA, ACE, ABC} con alfabeto {A, B, C, D, E}

6) Trovare codifica con algoritmo hi Huffman date le frequenze {A:44, B:53, C:12, D:10, E:15, F:3, G:55}

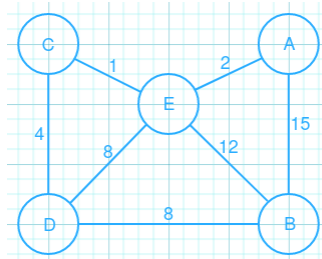
7) Complessità algoritmo Dijkstra su grafo con n nodi e m archi

- $O(n + m)$
- $O(m \log n)$
- $O(n * m)$
- $O(n \log n + m \log n)$

8) Quale è il caso migliore per l'algoritmo selectionsort?

- array vuoto
- array di liste
- liste di array
- nessuna delle altre

9) Trovare la lunghezza (la somma dei pesi degli archi) del minimo albero di copertura del seguente grafo (o comunque di uno simile)



- 13
- 18
- 20
- nessuna delle altre

10) Trovare la lunghezza della PLSC tra xyzzyx e xxyzxy

- 3
- 4
- 5
- nessuna delle altre

11) La visita preorder di un albero binario ha complessità:

- logaritmica sull'altezza
- lineare sul numero di nodi
- logaritmica sul numero di nodo
- lineare sull'altezza

12) Complessità nel caso medio per il quicksort?

- $O(n \log n)$
- $O(n)$
- $O(n^2)$
- $O(n^3)$

Risposte

1) **1 9 1 7 2 9** (classi con parametri diversi si comportano come classi diverse)

2) **errore** (viene lanciato un char, ma i catch catturano solo int)

3) **Si crea un figlio destro di M** (si scende per l'albero andando a destra se il valore è maggiore dell'etichetta del nodo considerato, altrimenti a sinistra, quando si raggiunge una foglia lo si inserisce come suo figlio)

4) **H F C D G M L V S**

5) **EBA, BBA, CEB, BEC, ABC, ACE**

6) Applicando l'algoritmo si ottiene **A:01 B:10 C:0001 D:00001 E:001 F:00000 G:11** (o altri equivalenti). Nel pretest si poteva osservare che in tutte le opzioni alcune lettere con frequenza alta avevano codifiche più lunghe di alcune con frequenza bassa, quindi si potevano tutte escludere e la risposta era **nessuna delle altre**

7) **$O(n \log n + m \log n)$**

8) **nessuna delle altre**

9) **nessuna delle altre** (con kruskal si vede che il minimo è 15 con il cammino A-E-C-D-B)

10) **4** (applicando l'algoritmo, o anche a occhio, si vede che la più lunga è xyzx)

11) **lineare sul numero di nodi**

12) **$O(n \log n)$**