

# Algoritmi e Strutture Dati

a.a. 2023/24

## Compito del 29/05/2024

Cognome: \_\_\_\_\_

Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_

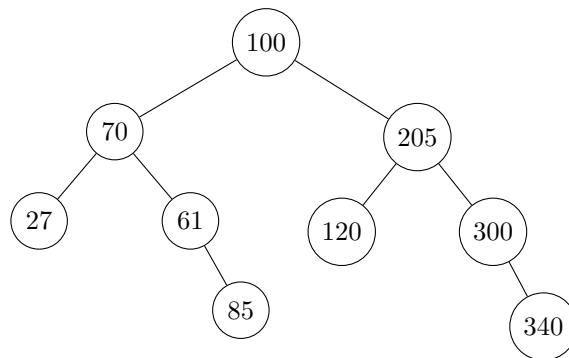
E-mail: \_\_\_\_\_

### Parte I

(30 minuti; ogni esercizio vale 2 punti)

**Avvertenza:** Si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Dato il seguente albero



- a) Eseguire una visita in preordine, una visita in ordine simmetrico e una visita in postordine elencando nei tre casi la sequenza dei nodi incontrati.  
b) L'albero è un albero binario di ricerca? Giustificare la risposta.

2. La Prof.ssa Raffaetà sostiene di aver sviluppato un algoritmo di complessità

$$T(n) = 7T(n/2) + n^2$$

per determinare tutti i cammini minimi in un grafo orientato e pesato, dove  $n$  rappresenta il numero di nodi del grafo in ingresso. Si dica se l'algoritmo in questione è preferibile o meno all'algoritmo di Floyd-Warshall e perché.

3. Il Prof. Pelillo sostiene di aver sviluppato un algoritmo di complessità

$$T(n) = 8T(n/2) + n^3$$

per determinare una clique massima in un grafo, dove  $n$  rappresenta il numero di nodi del grafo in ingresso. È plausibile che l'algoritmo in questione sia corretto? Perché?

# Algoritmi e Strutture Dati

a.a. 2023/24

## Compito del 29/05/2024

Cognome: \_\_\_\_\_

Nome: \_\_\_\_\_

Matricola: \_\_\_\_\_

E-mail: \_\_\_\_\_

### Parte II

(2.5 ore; ogni esercizio vale 6 punti)

**Avvertenza:** Si giustificino tecnicamente tutte le risposte. In caso di discussioni poco formali o approssimative gli esercizi non verranno valutati pienamente.

1. Sia `arr` un vettore di interi di dimensione  $n$ . Si assuma che nel vettore `arr` siano stati inseriti numeri interi positivi provenienti da un **albero binario di ricerca completo**  $T$ . In particolare, gli elementi di  $T$  sono stati inseriti in `arr` usando la stessa convenzione che si usa normalmente per la memorizzazione di uno heap in un vettore.
  - a) Scrivere una funzione **efficiente** `int maxBST(vector<int>& arr)` che restituisce il massimo di `arr`.
  - b) Siano `arr1` ed `arr2` due vettori che memorizzano due **alberi binari di ricerca completi**  $T1$  e  $T2$  aventi ciascuno  $n$  elementi, con le convenzioni sopra fissate. Sia  $k$  una chiave intera tale che tutte le chiavi di  $T1$  sono minori di  $k$  e tutte le chiavi di  $T2$  sono maggiori di  $k$ . Sia `arr` un vettore di dimensione  $2n + 1$ . Si vuole memorizzare in `arr` l'**albero binario di ricerca completo**  $T$  che si otterrebbe dalla fusione di  $T1$ ,  $k$ , e  $T2$ . Scrivere una funzione **efficiente** che restituisce `arr`. La funzione ha il seguente prototipo:  
`vector<int> mergeBST(vector<int>& arr1, vector<int>& arr2, int val)`
  - c) Determinare e giustificare la complessità delle funzioni `maxBST` e `mergeBST`.

Le funzioni devono essere in C o in C++ specificando il linguaggio scelto.

2. Per iscriversi agli esami gli studenti devono usare il numero di matricola che è rappresentato da una stringa di 7 caratteri con il seguente formato "S#####", dove "#" indica una cifra. Scrivere una procedura **efficiente** per ordinare in modo **decrescente** l'elenco degli iscritti a un esame in base al numero di matricola.  
Il prototipo della procedura è:  
`void riordina(vector<string>& arr)`

**Valutare e giustificare** la complessità della procedura proposta.

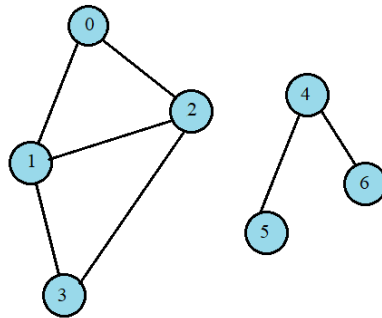
3. Si consideri il seguente algoritmo, che accetta in ingresso un grafo non orientato  $G = (V, E)$ :

```
MyAlgorithm(G)
1.  A =  $\emptyset$ 
2.  for each vertex  $u \in V[G]$  do
3.    MAKE-SET( $u$ )
4.  for each edge  $(u,v) \in E[G]$  do
5.    if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ ) then
6.      UNION( $u,v$ )
7.      A = A  $\cup$   $\{(u,v)\}$ 
8.  if  $|A| = n - 1$  then
9.    return TRUE
10. else
11.   return FALSE
```

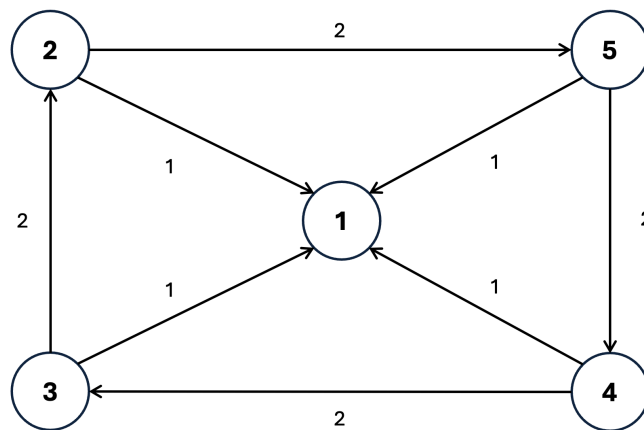
dove  $|A|$  denota il numero di elementi (cardinalità) dell'insieme  $A$ .

In quale caso `MyAlgorithm` restituisce TRUE? In questo caso, cosa conterrà l'insieme  $A$  alla fine dell'algoritmo? (Giustificare formalmente le risposte.)

Si simuli inoltre accuratamente l'esecuzione dell'algoritmo sul grafo seguente, mostrando l'evoluzione delle strutture dati coinvolte passo dopo passo:



4. Si scriva l'algoritmo di Dijkstra, si derivi la sua complessità, si dimostri la sua correttezza e si simuli la sua esecuzione sul seguente grafo **utilizzando il vertice 1 come sorgente**:



In particolare:

- a) si indichi l'ordine con cui vengono estratti i vertici  
b) si riempia la tabella seguente con i valori dei vettori  $d$  e  $\pi$ , iterazione per iterazione:

[illegible]