

# Algoritmi e Strutture Dati

a.a. 2023/24

## Second prova intermedia del 29/05/2024

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

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

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

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

1. 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.

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

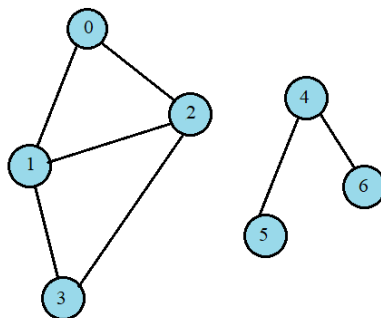
MyAlgorithm( $G$ )

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

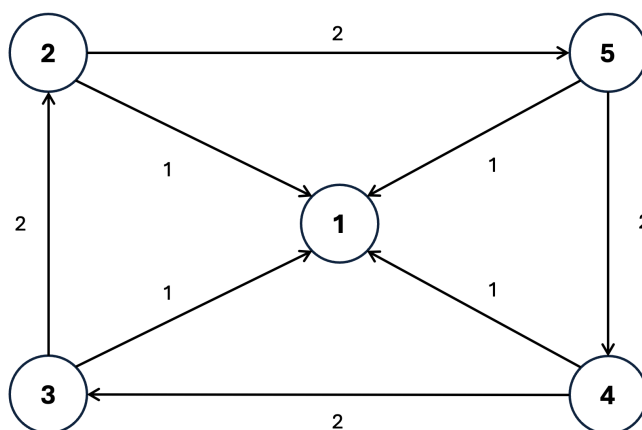
dove  $|A|$  denota il numero di elementi (cardinalità) dell'insieme  $A$ .

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:



3. 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:

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

[illegible]