

# Algoritmi e Strutture Dati

a.a. 2021/22

Compito del 12/9/2022

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. Sia  $H$  un **min-heap** contenente  $n$  interi **distinti** ed implementato con un vettore (come visto durante il corso). Rispondere alle seguenti domande, **motivando le risposte**.
  - a. Assumendo che i nodi interni di  $H$  siano già ordinati, è possibile ordinare tutti gli elementi di  $H$  con complessità strettamente inferiore a  $O(n \log n)$ ?
  - b. Sia  $k$  costante rispetto ad  $n$ . Quanto costa determinare la  $k$ -esima chiave più piccola di  $H$ ?
  - c. Quanto costa eliminare la chiave  $H[i]$ ?
2. Il Prof. Cook sostiene che, sebbene (come visto a lezione)  $3\text{-SAT-FNC} \leq_P \text{CLIQUE}$ , non sia vero il contrario, ovvero che  $\text{CLIQUE}$  non è riducibile polinomialmente a  $3\text{-SAT-FNC}$ . L'affermazione è corretta? (Spiegare.)
3. Dato un grafo orientato  $G$  con  $n$  vertici ed  $m$  archi, quante sono *esattamente* le operazioni di rilassamento ("relax") realizzate dall'algoritmo di Bellman-Ford su  $G$ ? Potremmo affermare qualcosa riguardo alla correttezza dell'algoritmo nel caso si facesse qualche "relax" in più? (Spiegare.)

# Algoritmi e Strutture Dati

a.a. 2021/22

Compito del 12/9/2022

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.

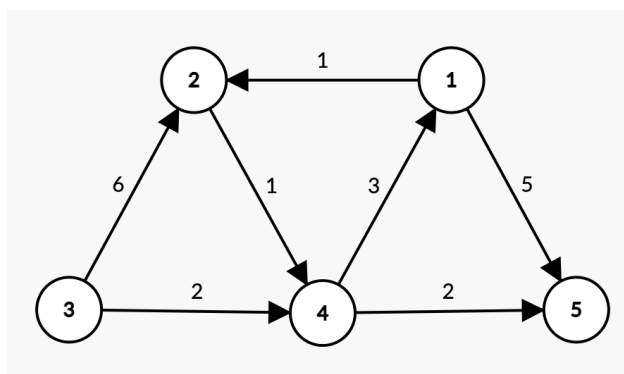
1. Sia  $T$  un albero generale i cui nodi hanno campi: **key**, **left-child** e **right-sib**. Scrivere una funzione **efficiente** in C o C++ che calcoli il numero di foglie di  $T$  e analizzarne la **complessità**.  
**Specificare quale linguaggio è stato utilizzato.**

2. Sia  $A$  un array di  $n$  numeri naturali. Si consideri il problema di stampare in ordine **crescente** tutti i numeri che compaiono in  $A$  almeno  $\lceil n/k \rceil$  volte, dove  $k > 0$  è una costante.

Si scriva una procedura **efficiente** che, dati  $A$ ,  $n$  e  $k$ , risolva il problema proposto.  
Valutare e giustificare la complessità dell'algoritmo proposto.

**Si devono scrivere le eventuali funzioni/procedure ausiliarie utilizzate.**

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:

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

	vertice 1		vertice 2		vertice 3		vertice 4		vertice 5	
	d[1]	$\pi$ [1]	d[2]	$\pi$ [2]	d[3]	$\pi$ [3]	d[4]	$\pi$ [4]	d[5]	$\pi$ [5]
dopo inizializzazione										
iterazione 1										
iterazione 2										
iterazione 3										
iterazione 4										
iterazione 5										

4. Sia  $G = (V, E)$  un grafo non orientato e connesso con funzione peso  $w : E \rightarrow \mathbb{R}$  e sia  $S$  un sottoinsieme di vertici di  $G$ . Si supponga che il taglio  $(S, V \setminus S)$  sia attraversato da un unico arco leggero  $(u, v)$  (ovvero:  $w(u, v) < w(x, y)$  per ogni altro arco  $(x, y)$  che attraversa il taglio).

Stabilire se le seguenti affermazioni sono vere o false:

- $G$  contiene un unico albero di copertura minimo
- Tutti gli alberi di copertura minimi di  $G$  contengono l'arco  $(u, v)$
- Esiste un albero di copertura minimo di  $G$  che contiene l'arco  $(u, v)$

Nel primo caso si offra una dimostrazione, nel secondo un controesempio.