

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 binario di ricerca** contenente n chiavi intere **distinte**. Sia k una chiave di T . Si consideri il problema di eliminare da T tutte le chiavi maggiori di k .
 - a) Si scriva una procedura in C **efficiente** per risolvere il problema proposto.
 - b) Calcolarne la complessità.
 - c) Scrivere i tipi in C Tree e Node.

Il prototipo della procedura è `void eliminaMaggioriK(Tree t, int k)`

2. Dimostrare come ordinare n numeri interi compresi nell'intervallo da 0 a $n^4 - 1$ nel tempo $O(n)$. Scrivere una funzione `ordina` che implementi tale ordinamento, ordinando gli n numeri in modo **non crescente**. Che tipo di algoritmo di ordinamento è stato utilizzato?
3. Si stabilisca quale problema risolve il seguente algoritmo, che accetta in ingresso un grafo orientato $G = (V, E)$ e la sua funzione peso $w : E \rightarrow \mathbb{R}$ (si assuma che G non contenga cappi e che $w(u,v) \geq 0$ per ogni $(u,v) \in E$):

`MyAlgorithm(G, w)`

```
1. m = +∞
2. for each u ∈ V[G]
3.   m = min {m, MyFunction(G, w, u)}
4. return m
```

`MyFunction(G, w, x)`

```
1. for each u ∈ V[G] \ {x}
2.   d[u] = +∞
3. d[x] = 0
4. Q = V[G]
```

```
5. for i = 1 to |V[G]|
```

```
6.   u = ExtractMin(Q)
```

```
7.   for each v ∈ V[G] \ Q
```

```
8.     if (u,v) ∈ E[G] then
```

```
9.       d[v] = min {d[v], d[u] + w(u,v)}
```

```
10.  a = +∞
```

```
11. for each u ∈ V[G] \ {x}
```

```
12.   if (u,x) ∈ E[G] then
```

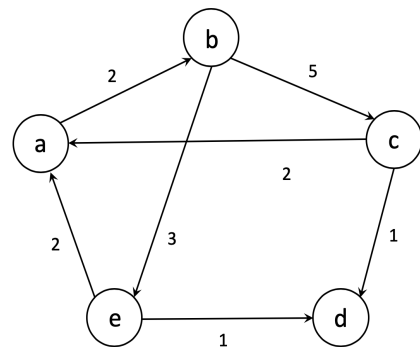
```
13.     a = min {a, d[u] + w(u,x)}
```

```
14. return a
```

Si dimostri la correttezza dell'algoritmo e si determini la sua complessità computazionale. (Nel caso in cui si volessero utilizzare risultati di correttezza di algoritmi noti, *non è necessario* fornire le dimostrazioni corrispondenti.)

Cosa restituisce l'algoritmo in presenza del grafo rappresentato sulla destra? Perché?

Si dica inoltre cosa restituisce `MyFunction` in corrispondenza dei cinque vertici del grafo.



4. Il seguente algoritmo accetta in ingresso un grafo non orientato $G = (V, E)$:

`MyAlgorithm(G)`

```
1. n = 0
```

```
2. for each u ∈ V[G]
```

```
3.   MAKE-SET(u)
```

```
4.   n = n + 1
```

```
5. for each (u,v) ∈ E[G]
```

```
6.   if FIND-SET(u) ≠ FIND-SET(v) then
```

```
7.     UNION(u,v)
```

```
8.     n = n - 1
```

```
9. return n
```

Cosa rappresenta il valore che `MyAlgorithm` restituisce al chiamante? Perché?

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

