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

1. Si consideri la struttura dati Min-Heap implementata tramite un vettore (secondo lo schema visto a lezione).
   a) Si scriva la definizione di Min-Heap.
   b) Si realizzi in modo **efficiente** la funzione Intersect(H1,H2) che dati due Min-Heap H1 e H2 contenenti rispettivamente n1 e n2 interi positivi, ritorna in output un nuovo Min-Heap contenente tutti e soli gli elementi che appartengono sia a H1 che a H2.
   c) Si determini e giustifichi la complessità in funzione di n1 e n2.

2. Sia A un vettore di interi **distinti** di dimensione n. L'ingegnere L.B. ci fornisce un algoritmo QuickSelect(A, p, q, r), dove p, q, r sono tali che $1 \le p \le r \le q \le n$.
   Dopo l'esecuzione di QuickSelect(A, p, q, r) valgono le seguenti condizioni su A:
   - A[r] contiene l'elemento che si troverebbe in posizione r in A se il sottovettore A[p..q] fosse ordinato;
   - A[p..q] è partizionato rispetto al pivot A[r], precisamente:
     A[i] < A[r] per ogni i tale che $p \le i < r$
     A[r] < A[j] per ogni j tale che $r < j \le q$.

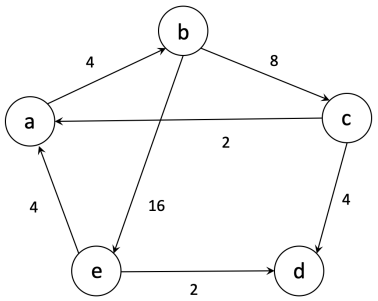
   L.B. ci dice che QuickSelect(A, p, q, r) ha complessità O(log m), dove $m = q - p + 1$.

   a) Si utilizzi QuickSelect per costruire un algoritmo **efficiente** per ordinare A. Si scriva lo pseudo-codice e si calcoli la complessità dell'algoritmo proposto.
   b) Può esistere un algoritmo QuickSelect con le proprietà sopra descritte? Motivare la risposta.

3. Si stabilisca quale problema risolve il seguente algoritmo, che accetta in ingresso un grafo orientato $G = (V, E)$, la sua funzione peso $w : E \to \mathrm{R}$ ed un vertice sorgente s (si assuma che G non contenga cappi e che $w(u,v) > 0$ per ogni $(u,v) \in E$). (Nell'algoritmo Q rappresenta una coda con priorità con campo chiave d.)

```
MyAlgorithm(G, w, s)
1.  for each u ∈ V[G] \ {s}
2.     d[u] = +∞
3.  d[s] = 0
4.  Q = V[G]
5.  for i = 1 to |V[G]|- 1
6.     u = ExtractMin(Q)
7.     for each v ∈ V[G]
8.        if (u,v) ∈ E[G] then
9.            d[v] = min {d[v], d[u] + log2 w(u,v)}
10.  a = +∞
11.  for each u ∈ V[G] \ {s}
12.     if (u,s) ∈ E[G] then
13.        a = min {a, d[u] + log2 w(u,s)}
14.  return 2^a
```

Si dimostri la correttezza dell'algoritmo e si determini la sua complessità computazionale assumendo che la coda con priorità Q sia implementata mediante un array lineare. Si dica inoltre cosa restituisce MyAlgorithm in presenza del seguente grafo, utilizzando i suoi cinque vertici come sorgente. Perché?



4. Il Prof. H. A. Milton sostiene di aver sviluppato un algoritmo di complessità

$$T(n) = 3T\left(\frac{n}{2}\right) + n^2$$

che riceve in ingresso un grafo non orientato G con n vertici e risponde TRUE se esiste in G un ciclo che passa per tutti i suoi vertici, e FALSE in caso contrario. Si dica, **giustificando tecnicamente la risposta**, se l'affermazione è verosimile.