

Algoritmi e Strutture Dati

a.a. 2014/15

Seconda Prova Intermedia del 27/05/2015

Cognome: _____

Nome: _____

Matricola: _____

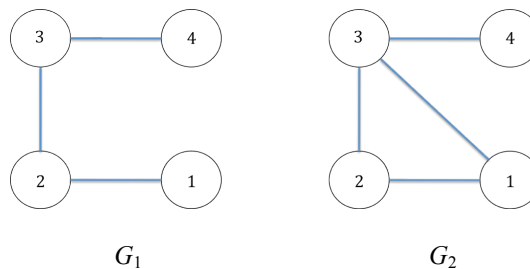
E-mail: _____

1. Scrivere l'algoritmo COUNTING-SORT per ordinare in senso **decrescente** n elementi di input, specificando le condizioni che devono essere soddisfatte per applicare tale algoritmo. Inoltre fare un'analisi della complessità dell'algoritmo, mettendo in evidenza quando risulta essere efficiente.
2. Il seguente algoritmo accetta in ingresso la matrice di adiacenza di un grafo non orientato e restituisce un valore Booleano (TRUE / FALSE)

```
MyAlgorithm( A )
1.  n = rows(A)           /* determina il numero di vertici del grafo */
2.  let B an n x n matrix /* alloca memoria per una matrice B n x n */
3.  for i = 1 to n
4.    for j = 1 to n
5.      b[i,j] = 0
6.      for k = 1 to n
7.        b[i,j] = b[i,j] + a[i,k]*a[k,j]
8.  for i = 1 to n
9.    for j = i + 1 to n
10.     if a[i,j]*b[i,j] <> 0 then
11.       return FALSE
12.  return TRUE
```

Si calcoli la complessità computazionale di MyAlgorithm e si determini la sua funzione (in quali casi restituisce TRUE? in quali casi FALSE?). (Nota: Nel determinare la complessità si ignori per comodità la complessità delle istruzioni 1 e 2).

Si simuli inoltre il suo comportamento sui seguenti due grafi, verificando che restituisca il risultato atteso:



3. Sia $G = (V, E)$ un grafo non orientato, connesso e pesato e sia $(S, V \setminus S)$ un taglio di G . Si supponga che l'arco leggero che attraversa il taglio sia unico e lo si denoti con (u, v) . In altri termini, per tutti gli altri archi (x, y) che attraversano il taglio, si avrà $w(u, v) < w(x, y)$. Si dimostri che *tutti* gli alberi di copertura minima di G dovranno necessariamente contenere l'arco (u, v) .