

Prova d'esame di Algoritmi e Strutture Dati  
21 gennaio 2026

Modalità d'esame:

1. prova intermedia (35%) + esercizi 3 e 4 (35%) + progetto (30%): 1 ora;
2. esercizi 1-4 (70%) + progetto (30%): 2 ore;

Punteggio massimo: 32 (8 per esercizio). Lode: > 30.

## Esercizio 1

1. Progettare un algoritmo **ricorsivo** con segnatura:

$$\minFigli(\text{Albero } T, \text{Intero } n) \rightarrow \text{Boolean}$$

che, preso in input un albero  $T$  e un intero  $n$ , restituisce il valore *true* se e solo se tutti i nodi dell'albero  $T$  hanno almeno  $n$  figli. Se  $T$  è l'albero vuoto, l'algoritmo deve restituire il valore *true*.

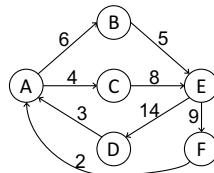
2. Illustrare il costo temporale dell'algoritmo proposto.

## Esercizio 2

1. Risolvere la seguente equazione di ricorrenza:  $T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 4T\left(\frac{n}{2}\right) + n^2, & \text{altrimenti} \end{cases}$
2. Risolvere la seguente equazione di ricorrenza usando il **metodo dell'iterazione**:  
$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ T\left(\frac{n}{2}\right) + 1, & \text{altrimenti} \end{cases}$$

## Esercizio 3

1. Riportare lo pseudo-codice dell'algoritmo di Dijkstra.
2. Illustrare l'esecuzione dell'algoritmo quando eseguito sul seguente grafo, prendendo come sorgente il nodo  $E$  e mostrando l'output restituito dall'algoritmo:



## Esercizio 4

1. Fornire la definizione di *minimo albero ricoprente* e discutere se sia possibile che un grafo ammetta più di un minimo albero ricoprente.
2. Mostrare l'algoritmo (pseudocodice) di Kruskal per il calcolo del minimo albero ricoprente.
3. Illustrare il costo temporale dell'algoritmo riportato, discutendo come si è pervenuti al risultato.
4. Mostrare l'esecuzione dell'algoritmo sul grafo in figura:

