

# Analisi Tematica e Frequenza Domande Orali

Domande organizzate per argomento e con un'indicazione della loro frequenza (numero approssimativo di occorrenze nel file).

## PARTE 1: TEORIA (Prof. Camurati)

### 1. Alberi Binari di Ricerca (BST) e Heap

- BST (Definizioni, proprietà, cancellazione, rotazioni): 40+
- Heap (Definizioni, proprietà, Heapify, HeapBuild): 35+
- Rango e BSTselect: 20
- Funzioni Tail-Recursive / Stack Frame: 26
- Heapify (Condizioni e Complessità): 21
- IBST (Interval BST): 16
- BST Partition: 11
- HeapBuild (Complessità): 11

### 2. Grafi: Visite, Componenti Connesse e DAG

- Componenti Connesse (CC) e Fortemente Connesse (SCC): 15
- Visite (DFS/BFS) (Complessità, strutture dati): 12
- Tipi di Archi (T, B, F, C): 10
- Kernel DAG: 9
- Punti di Articolazione e Bridge: 8
- Algoritmo di Kosaraju: 6

### 3. Tabelle di Hash

- Gestione Collisioni (Tutti i metodi): 30
- Cancellazione (specialmente in Open Addressing): 23
- Tabelle ad Accesso Diretto: 10
- Fattore di Carico (alpha): 9
- Funzioni di Hash (per stringhe, ecc.): 7
- Clustering: 5

### 4. Paradigmi Algoritmici e Complessità

- Programmazione Dinamica (DP): 34
- Paradigma Greedy: 24
- Divide and Conquer (Equazione generica): 15
- Memoization: 9
- Algoritmo di Karatsuba: 8

- Decrease and Conquer (Equazione generica): 7

## 5. Alberi Ricoprenti Minimi (MST)

- Definizione di MST: 15
- Teorema/Corollario sugli Archi Sicuri: 13
- ADT Union-Find: 10
- Algoritmo di Kruskal: 7
- Algoritmo di Prim: 5

## 6. Cammini Minimi

- Algoritmo di Bellman-Ford: 28
- Algoritmo di Dijkstra: 19
- Relaxation (Rilassamento): 14

## 7. Code a Priorità (PQueue)

- Implementazioni e Complessità (Generale): 26
- PQchange (Complessità e ottimizzazione): 14

## 8. Algoritmi di Ordinamento

- Quicksort (Caso peggiore, pivot, equazione): 32
- Merge Sort (Complessità, stabilità, equazione): 19
- Stabilità / "In loco": 15
- Limite Inferiore Complessità ( $O(N \log N)$ ): 9
- Altri (Insertion, Selection, Counting): 8

## 9. Teoria di Base e Calcolo Combinatorio

- Codici liberi da prefisso (Huffman): 40
- Powerset: 18
- Partizioni di un insieme (Algoritmo di Er): 14

## 10. Grafi: Definizioni e Problemi NP

- Cammino di Eulero / Hamilton: 28
- Grafo Completo (Definizione, numero archi): 20
- Cammini Massimi (Perché NP, come su DAG): 16
- Grafo Connesso (Definizione): 15
- Classi P, NP, NP-C: 11

# PARTE 2: PROGRAMMAZIONE (Prof. Cabodi)

## 1. Liste, Stack e Code

- Manipolazione Liste (Merge, Inversione, Cancellazione, ...): 95+
- Implementazione Stack/Code (FIFOinsert, STACKpush, ...): 32

## 2. Calcolo Combinatorio e Stringhe

- Problemi risolvibili con Modelli Combinatori (Disposizioni, ...): 100+
- Manipolazione di Stringhe e Vettori: 70+
- Implementazione Algoritmi (Sort, Search, ...): 20
- Implementazione Codici di Huffman: 5

## 3. Alberi Binari di Ricerca (BST)

- Funzioni di Conteggio/Analisi (nodi a prof. k, in intervallo, ...): 30
- Inserimento (in foglia, in radice): 16
- Funzioni di Verifica (isomorfo, bilanciato, completo, ...): 15
- Funzioni di Ricerca/Navigazione (Successore, Predecessore): 13
- Calcolo Profondità / Altezza: 13
- Cancellazione / Manipolazione: 11
- BSTselect (con/senza rango): 6
- Duplicare un BST (BSTdup): 3

## 4. Grafi

- Ricerca/Conteggio Vertici/Archi (grado max, raggiungibili in k passi, ...): 55+
- Verifica Cammini/Cicli (se un array è un cammino, contare triangoli, ...): 45+
- Implementazione Visite (BFS/DFS): 40+
- Verifica Proprietà (completo, DAG, clique, punto articolazione, ...): 30
- Manipolazione (rimuovere vertice/arco, trasposizione, ...): 16

## 5. Heap e Code a Priorità (PQueue)

- Implementazione Operazioni (Insert, Extract, Change): 40+
- Manipolazione (Trasformare vettore in albero): 3

## 6. Tabelle di Hash

- Implementazione Operazioni (Insert/Delete, chaining/probing): 26
- Analisi (Calcolo lunghezza media liste): 2

# RIEPILOGO ARGOMENTI PRINCIPALI (Ordinati per Frequenza Totale)

1. **Calcolo Combinatorio e Stringhe (Programmazione):** (100+) Domande di programmazione che richiedono l'uso di disposizioni, permutazioni, ecc., spesso con pruning, o manipolazione di stringhe.
2. **Liste, Stack e Code (Programmazione):** (95+) Esercizi di manipolazione di liste (merge, inversione, cancellazione) e implementazioni di stack/code.
3. **Grafi (Programmazione):** (55+) Conteggio e ricerca di vertici/archi con varie proprietà (grado, raggiungibilità, ecc.).
4. **Grafi (Programmazione):** (45+) Verifica di cammini, cicli e conteggio di triangoli.

5. **BST e Heap (Teoria):** (40+) Definizioni, proprietà, cancellazione, rotazioni.
6. **Codici liberi da prefisso (Huffman) (Teoria):** (40) Definizione, funzionamento, vantaggi.
7. **Heap e PQueue (Programmazione):** (40+) Implementazione di HeapInsert, PQextract, PQchange.
8. **Grafi (Programmazione):** (40+) Scrittura di visite (BFS/DFS) per risolvere problemi specifici.
9. **Heap (Teoria):** (35+) Definizione, proprietà, implementazione su vettore.
10. **Programmazione Dinamica (DP) (Teoria):** (34) Definizione, condizioni di applicabilità, esempi.
11. **Quicksort (Teoria):** (32) Caso peggiore, scelta del pivot, equazione alle ricorrenze.
12. **Liste e Stack (Programmazione):** (32) Implementazione di ADT Stack e Coda.
13. **BST (Programmazione):** (30) Esercizi di conteggio nodi (a profondità k, in un intervallo, foglie, ecc.).
14. **Grafi (Programmazione):** (30) Verifica di proprietà (DAG, completo, clique, punto di articolazione).
15. **Tabelle di Hash (Teoria):** (30) Gestione delle collisioni (chaining, probing).
16. **Cammino di Eulero / Hamilton (Teoria):** (28) Definizioni e differenze.
17. **Algoritmo di Bellman-Ford (Teoria):** (28) Funzionamento, DP, cicli negativi.
18. **Funzioni Tail-Recursive / Stack Frame (Teoria):** (26) Definizioni, differenze, gestione memoria.
19. **Code a Priorità (PQueue) (Teoria):** (26) Implementazioni e complessità.
20. **Tabelle di Hash (Programmazione):** (26) Implementazione di STinsert e STdelete.
21. **Paradigma Greedy (Teoria):** (24) Definizione, esempi (Kruskal, Prim, Dijkstra).
22. **Tabelle di Hash (Teoria):** (23) Gestione della cancellazione in open addressing.
23. **Heapify (Teoria):** (21) Condizioni di applicabilità e complessità.
24. **Rango e BSTselect (Teoria):** (20) Definizione di rango, BSTselect con e senza info N.
25. **Grafo Completo (Teoria):** (20) Definizione e numero di archi.
26. **Implementazione Algoritmi (Programmazione):** (20) Scrittura di Sort, Search, ecc.
27. **Merge Sort (Teoria):** (19) Complessità, stabilità, equazione.
28. **Algoritmo di Dijkstra (Teoria):** (19) Paradigma, strutture dati, condizioni.
29. **Powerset (Teoria):** (18) Definizione e algoritmi per generarli.
30. **Cammini Massimi (Teoria):** (16) Problema NP, soluzione su DAG.
31. **IBST (Interval BST) (Teoria):** (16) Definizione e funzionamento.
32. **BST (Programmazione):** (16) Inserimento (in foglia e in radice).
33. **Grafi (Programmazione):** (16) Manipolazione (rimozione vertici/archi, trasposto).
34. **Definizione di MST (Teoria):** (15) Definizione e proprietà.
35. **Divide and Conquer (Teoria):** (15) Equazione alle ricorrenze generica.
36. **Grafo Connesso (Teoria):** (15) Definizione.
37. **Stabilità / "In loco" (Teoria):** (15) Definizioni per algoritmi di ordinamento.
38. **BST (Programmazione):** (15) Funzioni di verifica (isomorfo, bilanciato, completo).
39. **PQchange (Teoria):** (14) Complessità e ottimizzazione (con qp).
40. **Partizioni di un insieme (Algoritmo di Er) (Teoria):** (14) Definizione e algoritmo.

41. **Relaxation (Rilassamento) (Teoria):** (14) Definizione del concetto.
42. **Teorema/Corollario Archi Sicuri (Teoria):** (13) Definizioni (taglio, arco sicuro).
43. **BST (Programmazione):** (13) Ricerca di successore e predecessore.
44. **BST (Programmazione):** (13) Calcolo di profondità e altezza.
45. **Visite (DFS/BFS) (Teoria):** (12) Complessità e strutture dati.
46. **BST Partition (Teoria):** (11) Funzionamento e utilizzo.
47. **HeapBuild (Teoria):** (11) Complessità  $O(n)$  e perché.
48. **Classi P, NP, NP-C (Teoria):** (11) Definizioni e relazioni.
49. **BST (Programmazione):** (11) Cancellazione e manipolazione nodi.
50. **ADT Union-Find (Teoria):** (10) Funzionamento (spesso con Kruskal).
51. **Tipi di Archi (T, B, F, C) (Teoria):** (10) Come identificarli nella DFS.
52. **Tabelle Accesso Diretto (Teoria):** (10) Definizioni e complessità.
53. **Limite Inferiore Complessità Ordinamento (Teoria):** (9) Dimostrazione  $O(N \log N)$ .
54. **Kernel DAG (Teoria):** (9) Definizione e legame con SCC.
55. **Fattore di Carico (Teoria):** (9) Definizione e valori (chaining vs open addressing).
56. **Memoization (Teoria):** (9) Definizione e differenza con DP.
57. **Punti di Articolazione / Bridge (Teoria):** (8) Definizione e come trovarli.
58. **Algoritmo di Karatsuba (Teoria):** (8) Vantaggio (3 prodotti vs 4).
59. **Altri Algoritmi di Ordinamento (Teoria):** (8) Counting Sort, Insertion Sort, ecc.
60. **Algoritmo di Kruskal (Teoria):** (7) Paradigma e funzionamento.
61. **Decrease and Conquer (Teoria):** (7) Equazione alle ricorrenze.
62. **Funzioni di Hash (Teoria):** (7) Come crearle (es. per stringhe).
63. **BSTselect (Programmazione):** (6) Implementazione con e senza rango.
64. **Algoritmo di Kosaraju (Teoria):** (6) Funzionamento.
65. **Algoritmo di Prim (Teoria):** (5) Paradigma e funzionamento.
66. **Clustering (Teoria):** (5) Definizione (in tabelle di hash).
67. **Huffman (Programmazione):** (5) Implementazione dell'algoritmo.
68. **BST (Programmazione):** (3) Duplicare un albero (BSTdup).
69. **Heap (Programmazione):** (3) Manipolazione (es. da vettore a albero).
70. **Tabelle di Hash (Programmazione):** (2) Calcolo lunghezza media liste.