

# Analisi degli Esami Passati: Algoritmi e Strutture Dati

Richard Feynman (Simulato)

3 luglio 2025

Ah, un'ottima strategia! Per vincere una sfida, devi prima capirla. Guardare i vecchi esami è come essere un fisico che studia i risultati di esperimenti passati. Non stai solo memorizzando le risposte; stai cercando gli schemi, i principi profondi che le domande cercano di sondare. Si tratta di capire il *carattere* del problema.

Ho dato un'occhiata approfondita a questa seconda parte dei tuoi esami, la sezione "Algoritmi e strutture dati". Riguarda interamente i meccanismi del calcolo. Non solo le idee astratte, ma i veri e propri bulloni di come costruiresti una macchina pensante, come organizzeresti le informazioni e come ne misureresti l'efficienza. Alcune idee spuntano fuori più e più volte, come temi ricorrenti in una grande storia. Se vuoi sapere su cosa scommettere, ecco cosa ho scoperto.

Basandomi sulla mia analisi di questi temi d'esame passati, gli argomenti che compaiono con la regolarità più sorprendente sono:

## 1. Il Racconto di Due Macchine: Turing vs. RAM

Questo è il pezzo grosso. È presente in quasi ogni singolo esame. Ne sono ossessionati! Gli esaminatori vogliono vedere che hai capito la differenza tra due modelli fondamentali di calcolo:

- **La Macchina di Turing (TM):** Pensa a questa come a un robot semplice e metodico con un lungo nastro di carta. Può solo leggere un simbolo alla volta, scriverne uno nuovo e spostare il nastro a sinistra o a destra [1, 2, 3]. È il modello più basilare, il fondamento. Ti chiedono di progettare queste macchine, sia con un singolo nastro sia a volte con nastri multipli (il che semplifica la vita) [4, 5], per compiti specifici come riconoscere un linguaggio o tradurre una stringa. La chiave qui è il processo meccanico, passo dopo passo, e la sua complessità, che è spesso frustrantemente lenta, come rimanere impantanati in un tempo quadratico ( $O(n^2)$ ) [6, 7].
- **La Macchina RAM:** Questa assomiglia di più a un computer moderno. Ha celle di memoria a cui puoi saltare direttamente. Niente più avanzamenti lenti su un nastro! Puoi prelevare dati da qualsiasi posizione in tempo costante. Ti chiederanno di risolvere lo *stesso* problema con una macchina RAM [8, 9, 10] per vedere se apprezzi la potenza e la velocità che guadagni.

### L'idea centrale

Vogliono che tu confronti e contrapponga. Che tu veda un problema e dica: "Con un semplice nastro, questa è una seccatura... dovrò scorrere avanti e indietro, perdendo un sacco di tempo. Ma con l'accesso casuale, posso semplicemente memorizzare i miei contatori in un paio di celle e farlo in un lampo!" Si tratta di capire i compromessi tra diverse architetture di macchine.

## 2. L'Arte dell'Organizzazione: Le Strutture Dati

Hai una montagna di dati. Come la sistemi in modo da poter trovare le cose rapidamente? Questo è l'intero gioco delle strutture dati. Le domande non sono solo “Cos'è una tabella hash?”. Sono “Ecco un problema. Inventi un sistema per risolverlo in modo efficiente”.

I personaggi più popolari in questa commedia sono:

- **Alberi, Alberi, Alberi:** Specialmente gli Alberi Binari di Ricerca (BST) [11] e i loro cugini più sofisticati e autobilanciati come gli Alberi Rosso-Neri [12]. Amano chiederti di eseguire operazioni su di essi: inserire, cancellare [13], invertire [14], o anche procedure ricorsive personalizzate come rimuovere tutte le foglie [15]. Un albero è un modo meraviglioso per imporre un ordine al caos.
- **Tabelle Hash:** Quando hai bisogno di qualcosa di veloce — intendo, veramente veloce, come un tempo costante in media — ti affidi a una tabella hash. Sono una soluzione ricorrente per problemi che richiedono ricerche, inserimenti o cancellazioni rapide, come contare le occorrenze di un elemento [16] o collegare il codice fiscale di una persona ai suoi dati [17].
- **Strutture Composte:** È qui che il gioco si fa divertente. Ti daranno una serie complessa di requisiti, come “cerca per data in tempo logaritmico, ma cerca per codice fiscale in tempo costante” [18]. Il trucco è capire che nessuna singola struttura basterà. Devi essere un ingegnere intelligente e combinarle! Ad esempio, potresti usare un albero ordinato per data, in cui ogni nodo ha anche un puntatore che è memorizzato in una tabella hash indicizzata per codice fiscale [17]. Si tratta di costruire una macchina migliore partendo da parti più semplici.

## 3. Lo Strumento Universale: Algoritmi su Array

Molti problemi si riducono a una lista di numeri in un array. Queste domande testano la tua capacità di pensare in modo algoritmico. Una soluzione ingenua potrebbe essere quella di confrontare ogni elemento con ogni altro, ma è lenta e inelegante ( $O(n^2)$ ). Lo schema qui è quasi sempre lo stesso:

1. **Ordinalo!** Ordinando prima l'array, ne riveli la struttura. Questo costa quasi sempre  $O(n \log n)$  [19, 20].
2. **Poi, scandiscilo in modo intelligente.** Dopo l'ordinamento, di solito puoi risolvere il problema con una singola passata lineare attraverso i dati, che costa solo  $O(n)$  [21].

Vedi questo schema per trovare la coppia di numeri più vicina tra due array [21], trovare la coppia di numeri identici più distanti [22], o per il problema “MAXPITAGORA” [23]. Il principio è semplice: investire un po' di tempo all'inizio per organizzare i dati può portare a enormi risparmi in seguito.

## 4. Quanto Velocemente Può Crescere? Le Equazioni di Ricorrenza

Questa è una questione più matematica, ma compare abbastanza spesso da essere importante [24, 25]. Ti daranno una formula ricorsiva, che descrive come il tempo di esecuzione di un algoritmo dipende dal tempo di esecuzione per input più piccoli, e tu devi capire la complessità generale. Lo strumento chiave qui è il **Teorema Master**, una potente scorciatoia per risolvere un'intera classe di questi problemi [26, 27]. È come avere una tavola degli integrali per risolvere le equazioni: se conosci gli schemi, puoi semplicemente scrivere la risposta.

Quindi, amico mio, non studiare solo una raccolta casuale di nozioni. Concentrati su questi principi fondamentali. Cerca di capire il *perché*. Perché una macchina RAM è più veloce? Perché ordinare aiuta? Perché dovresti combinare un albero e una tabella hash? Se riesci a spiegare queste cose a te stesso, in modo semplice e chiaro, come se le stessi spiegando a uno studente curioso, allora non solo supererai l'esame, ma capirai la materia. E questa è l'unica cosa che abbia mai avuto un valore.

## Riferimenti bibliografici

- [1] Esame del 7 Settembre 2023, Esercizio 3.
- [2] Esame del 10 Febbraio 2023, Esercizio 3.
- [3] Esame del 10 Gennaio 2023, Esercizio 3.
- [4] Esame del 30 Gennaio 2025, Esercizio 4.
- [5] Esame del 7 Luglio 2023, Esercizio 3.
- [6] Esame del 30 Gennaio 2025, Esercizio 3.
- [7] Esame del 7 Luglio 2023, Esercizio 3.
- [8] Esame del 30 Gennaio 2025, Esercizio 4.
- [9] Esame del 12 Febbraio 2024, Esercizio 3.
- [10] Esame del 4 Settembre 2024, Esercizio 3.
- [11] Esame del 15 Gennaio 2025, Esercizio 3.
- [12] Esame del 12 Febbraio 2024, Esercizio 4.
- [13] Esame del 15 Gennaio 2025, Esercizio 3.
- [14] Esame del 19 Giugno 2023, Esercizio 4.
- [15] Esame del 19 Giugno 2023, Esercizio 4.
- [16] Esame del 20 Giugno 2024, Esercizio 4.
- [17] Esame del 12 Febbraio 2024, Esercizio 4.
- [18] Esame del 12 Febbraio 2024, Esercizio 4.
- [19] Esame del 7 Settembre 2023, Esercizio 4.
- [20] Esame del 4 Settembre 2024, Esercizio 3.
- [21] Esame del 4 Settembre 2024, Esercizio 4.
- [22] Esame del 7 Settembre 2023, Esercizio 4.
- [23] Esame del 10 Gennaio 2023, Esercizio 4.
- [24] Esame del 19 Giugno 2023, Esercizio 3.
- [25] Esame del 20 Giugno 2024, Esercizio 3.
- [26] Esame del 19 Giugno 2023, Esercizio 3.
- [27] Esame del 20 Giugno 2024, Esercizio 3.