

ESAME DI ALGORITMI
Università degli Studi di Catania
Corso di Laurea Triennale in Informatica
27 ottobre 2025

Si risolvano i seguenti esercizi in un tempo non superiore a 3 ore.

1. Si considerino le due procedure **HalfSearchProperty** e **RBTreeSum** descritte di seguito. La prima opera su un *array ordinato* di n elementi, mentre la seconda su un *albero binario rosso-nero* con n nodi. Per ciascuna di esse, scrivere l'equazione di ricorrenza che descrive il tempo di esecuzione $T(n)$ e risolverla utilizzando uno dei metodi noti (albero di ricorsione, sostituzione o Teorema Master).

Pseudocodice A

```
1 HalfSearchProperty(A, n, s)
2   if n == 1 then
2     return (A[1] > 0)
4   e = RicercaBinaria(A, n, s)
5   if (A[n/2] > e) then
5     return HalfSearchProperty(A, n/2, s)
```

Pseudocodice B

```
1 RBTreeSum(x)
2   if size(x) == 1 then
2     return key(x)
4   somma := inOrderSum(x)
6   y := left(x)
7   return RBTreeSum(y)
```

dove la funzione **RicercaBinaria(A, n, s)** rappresenta una classica ricerca binaria su di un array di dimensione n , mentre la funzione **inOrderSum(x)** rappresenta una visita in-order dell'albero binario radicato in x .

2. Scrivere una funzione ricorsiva **checkBST(x)** che verifichi se l'albero binario radicato nel nodo x rispetta le proprietà di un BST. La funzione deve restituire **true** se per ogni nodo v dell'albero radicato in x tutti i valori nel sottoalbero sinistro sono minori di v , e tutti quelli nel sottoalbero destro sono maggiori.
3. Un sistema registra in tempo reale i punteggi dei giocatori di un videogioco online. Si vuole mantenere in ogni momento la Top-Ten dei giocatori con i punteggi più alti, aggiornandola in tempo efficiente man mano che arrivano nuovi risultati. Per ciascun giocatore si considera solo il suo punteggio migliore. Progettare un algoritmo e una struttura dati che permettano di aggiornare la classifica dopo ciascun nuovo punteggio. Descrivere l'idea di base e stimare la complessità dell'operazione di aggiornamento della struttura.
4. Si forniscano le funzioni ricorsive sulle quali si basano i due algoritmi per la risoluzione del problema dei cammini minimi tra tutte le coppie studiati a lezione: **AllPairsShortestPath** e **FloydWarshall**. Per ognuna delle funzioni ricorsive, fornire un'interpretazione dei parametri e delle variabili utilizzate al suo interno.
5. Si enunci il problema di ottimizzazione affrontato e risolto dal ben noto algoritmo di Huffman. Si fornisca la funzione che assegna ad un albero di Huffman T il suo costo. Si dimostri che il problema gode della proprietà della scelta greedy.
6. Fornire un esempio concreto di un albero rosso-nero valido contiene 10 chiavi distinte in cui un'operazione di cancellazione provoca la diminuzione dell'altezza nera dell'albero. Successivamente, fornire un'altro esempio concreto di un albero rosso-nero valido contiene 10 chiavi distinte in cui un'operazione di inserimento provoca l'aumento dell'altezza nera dell'albero. Per entrambi gli esempi, disegnare la configurazione dell'albero prima e dopo le operazioni..