

Algoritmi e Strutture Dati

18 giugno 2025

Note

1. La leggibilità è un prerequisito: parti difficili da leggere potranno essere ignorate.
2. Quando si presenta un algoritmo è fondamentale spiegare l'idea e motivarne la correttezza.
3. L'efficienza e l'aderenza alla traccia sono criteri di valutazione delle soluzioni proposte.
4. Si consegnano tutti i fogli, con nome, cognome, matricola e l'indicazione *bella copia* o *brutta copia*.

Domande

Domanda A (7 punti) Dare la definizione formale delle classi $O(f(n))$ e $\Omega(f(n))$ per una funzione $f(n)$. Mostrare che se $f(n) = O(n)$ e $g(n) = n^2 - f(n)$ allora $g(n) = \Omega(n^2)$.

Domanda B (6 punti) Indicare, in forma di albero binario, il codice prefisso ottenuto tramite l'algoritmo di Huffman per l'alfabeto $\{a, b, c, d, e, f\}$, supponendo che ogni simbolo appaia con le seguenti frequenze:

a	b	c	d	e	f
12	7	14	30	10	27

Spiegare brevemente il processo di costruzione del codice.

Esercizi

Esercizio 1 (9 punti) Realizzare un arricchimento degli alberi binari di ricerca che permetta di ottenere per ogni nodo x , in *tempo costante*, il numero delle foglie nel sottoalbero radicato in x .

Indicare quali campi occorre aggiungere ai nodi. Fornire il codice per la funzione `leaves(x)` che restituisce il numero delle foglie nel sottoalbero radicato in x e per la procedura di inserimento di un nodo `insert(T, z)`. Per entrambe valutare la complessità.

Esercizio 2 (10 punti) Abbiamo n programmi da eseguire sul nostro computer. Ogni programma j , dove $j \in \{1, 2, \dots, n\}$, ha lunghezza ℓ_j , che rappresenta la quantità di tempo richiesta per la sua esecuzione. Dato un ordine di esecuzione $\sigma = j_1, j_2, \dots, j_n$ dei programmi (cioè, una permutazione di $\{1, 2, \dots, n\}$), il tempo di completamento $C_{j_i}(\sigma)$ del j_i -esimo programma è dato quindi dalla somma delle lunghezze dei programmi $j_1, j_2, \dots, j_i$. L'obiettivo è trovare un ordine di esecuzione σ che minimizza la somma dei tempi di completamento di tutti i programmi, cioè $\sum_{j=1}^n C_j(\sigma)$.

- (a) Dare un semplice algoritmo greedy per questo problema, e valutarne la complessità.
- (b) Dimostrare la proprietà di scelta greedy dell'algoritmo del punto (a), cioè che esiste un ordine di esecuzione ottimo σ^* che contiene la scelta greedy.