

# Progettazione degli Algoritmi - Settembre 2022

13 Settembre, 2022

**Esercizio 1. Conteggio di monete:** Trovate un algoritmo che, dato un valore  $N \geq 0$ , conti in quanti modi diversi è possibile ottenere la somma totale  $N$  utilizzando monete di valore 1, 2, 5, 10. L'algoritmo deve avere complessità  $O(N)$ .

Ad esempio  $N = 7$  può essere ottenuto in 6 modi, come

$(1, 1, 1, 1, 1, 1, 1), (1, 1, 1, 1, 1, 2), (1, 1, 1, 2, 2), (1, 2, 2, 2), (1, 1, 5), (2, 5)$

**Motivare BENE la correttezza e la complessità dell'algoritmo proposto.**

**Esercizio 2. Stampa:** Dato un numero  $n \geq 0$  e un insieme  $I \subseteq \{0, \dots, n\}$  di numeri interi, definiamo  $S(n, I)$  l'insieme di stringhe binarie di lunghezza  $n$  per cui il numero di occorrenze della cifra 1 sia nell'insieme  $I$ . Ad esempio se  $n = 3$  e  $I = \{0, 1, 3\}$  allora l'insieme  $S(n, I)$  è costituito dalle stringhe

000, 001, 010, 100, 111 .

Trovate un algoritmo di complessità  $O(n \cdot |S(n, I)|)$  che dati  $n$  ed  $I$  stampi tutte le stringhe binarie in  $S(n, I)$  **in ordine lessicografico**. In particolare **non è sufficiente** stampare tutte le stringhe con uno specifico numero di 1, e ripetere il tutto per ogni valore in  $I$ . Questa procedura infatti non produrrebbe le stringhe in ordine lessicografico. Potete assumere che il parametro  $I$  sia dato come un array già ordinato in maniera crescente, privo di valori ripetuti, e contenente solo valori tra 0 e  $n$ .

**Suggerimento:** potreste tenere traccia dell'intervallo di valori in  $I$  che possono essere ancora ottenuti.

**Motivare BENE la correttezza e la complessità dell'algoritmo proposto, calcolando il numero di nodi interni e di foglie nell'albero di computazione, e mettendoli in corrispondenza con  $|S(n, I)|$ .**

**Esercizio 3. Grafi:** Considerate il grafo nella figura che segue, ed eseguite una visita in profondità a partire dal vertice 0. Durante la visita, nel caso l'algoritmo debba esplorare il vicinato di un vertice, lo farà sempre in ordine crescente rispetto all'indice. Disegnate l'albero di visita ottenuto da questa DFS, e descrivete gli archi di attraversamento, archi in avanti e gli archi all'indietro ottenuti da questa visita.

Considerate il grafo di partenza dal quale siano stati eliminati gli archi all'indietro: calcolate l'ordine topologico sul grafo ottenuto utilizzando l'algoritmo basato sulla visita in profondità.

**Motivare BENE la vostra risposta.**

