



Esame di Ammissione - A.A. 2025/26

Prova Scritta di Informatica

1. Permutazioni

Dato un array di elementi, si descriva un algoritmo per generarne tutte le possibili permutazioni.

- **Elementi Distinti:** Sviluppare un algoritmo che, dato un array in cui gli elementi sono tutti distinti, ne generi tutte le permutazioni, ogni permutazione deve essere generata una sola volta.
- **Elementi Ripetuti:** Adattare l'algoritmo precedente per gestire correttamente il caso in cui l'array di input possa contenere elementi duplicati.

2. Edit Distance

Date due stringhe A e B, calcolare la **distanza di Levenshtein**, ovvero il numero minimo di operazioni necessarie per trasformare la stringa A nella stringa B. Le operazioni possibili sono inserimento, cancellazione o sostituzione di un carattere. Ogni operazione ha un costo unitario.

- **Solo Sostituzione:** Risolvere il problema considerando unicamente l'operazione di **sostituzione** di un carattere.
- **Operazioni Complete:** Estendere la soluzione per includere anche le operazioni di **inserimento** e **cancellazione** di un carattere.
 - **Suggerimento:** Il problema può essere risolto calcolando il costo minimo per trasformare ogni prefisso di A in ogni prefisso di B.

3. Parentesi Bilanciate

Data una stringa contenente diverse tipologie di parentesi ($()$, $\{\}$, $[\]$), si descriva un algoritmo per verificare se le parentesi sono "bilanciate", ovvero se ogni parentesi aperta viene chiusa dalla corrispondente parentesi dello stesso tipo e nell'ordine corretto.

- **Tipo Singolo:** Risolvere il problema considerando solo le parentesi tonde $()$.
- **Tipi Multipli:** Estendere l'algoritmo per gestire parentesi tonde, quadre e graffe.
- **Carattere Jolly:** Modificare l'algoritmo per gestire anche la presenza di un carattere jolly ($*$), che può essere interpretato come una parentesi aperta, parentesi chiusa, o stringa vuota. L'obiettivo è determinare se esiste un'interpretazione dei caratteri jolly che renda la stringa valida.