Esercizio 1: Problema dello scheduling con profitti

Un'azienda deve eseguire n lavori su una singola macchina. Ogni lavoro j ha un tempo di esecuzione t[j] e un profitto p[j] se viene completato. I lavori possono essere eseguiti in qualsiasi ordine, ma una volta iniziato un lavoro deve essere completato senza interruzioni. L'obiettivo è massimizzare il profitto totale entro un tempo limite T.

i. Formalizzare la nozione di soluzione per il problema e il relativo profitto. Mostrare che vale la proprietà della sottostruttura ottima e individuare una scelta che gode della proprietà della scelta greedy.

ii. Sulla base della scelta greedy individuata al passo precedente, fornire un algoritmo greedy schedule(t, p, n, T) che, dati in input gli array dei tempi t[1..n] e dei profitti p[1..n], il numero di lavori n e il tempo limite T, restituisce una soluzione ottima.

iii. Valutare la complessità dell'algoritmo.

iv. Dimostrare la correttezza dell'algoritmo.

Esercizio 2: Problema del bilanciamento del carico

Si hanno m macchine e n lavori, dove ogni lavoro j richiede un tempo di esecuzione t[j]. Si vuole assegnare ogni lavoro a una macchina in modo da minimizzare il tempo di completamento massimo tra tutte le macchine.

i. Formalizzare la nozione di soluzione per il problema e la relativa funzione obiettivo da minimizzare. Proporre una strategia greedy per l'assegnazione dei lavori.

ii. Implementare un algoritmo greedy loadBalance(t, n, m) che, dato l'array dei tempi di esecuzione t[1..n], il numero di lavori n e il numero di macchine m, restituisce un'assegnazione dei lavori alle macchine.

iii. Analizzare la complessità dell'algoritmo proposto.