Sciatori

Siano dati:

- n sciatori di altezza $H[0], \ldots, H[n-1]$;
- n paia di sci di lunghezza $L[0], \ldots, L[n-1]$.

Problema: trovare un assegnamento sciatore-sci che minimizzi la somma delle differenze assolute fra l'altezza degli sciatori e la lunghezza degli sci a loro assegnati.

In altre parole, se allo sciatore i è assegnato il paio di sci $m(i)$ (matching), minimizzare la seguente quantità:

$$\sum_{i=0}^{n-1} |H[i] - L[m(i)]|$$

Sciatori

Soluzione proposta

Identifichiamo la coppia sciatore-sci la cui differenza è minore, assegniamo lo sci allo sciatore e ci riduciamo ad un problema più piccolo, con uno sciatore in meno e uno sci in meno.

Dimostrare la correttezza dell'algoritmo proposto oppure trovare un controesempio.

Sfilatino alla Nutella

Nella sagra di Hateville, è stato realizzato lo sfilatino alla Nutella più lungo del mondo, lungo L centimetri. Ora si tratta di realizzare un altro record: il maggior numero di persone servite con lo stesso sfilatino.

Alla sagra sono presenti n persone, dove la persona i-esima chiede un *segmento* di sfilatino lungo $V[i]$ centimetri; secondo il regolamento, ogni richiesta va servita esattamente, ovvero se la persona i verrà servita, riceverà il segmento richiesto. Tutte le lunghezze sono intere positive.

Scrivere un algoritmo che restituisca il numero massimo di persone che possono essere servite con lo sfilatino. Non è necessario utilizzare tutto lo sfilatino.

Oltre a calcolare la complessità dell'algoritmo proposto, discutere anche la correttezza, menzionando la tecnica scelta e specificando bene perché tale tecnica può essere applicata in questo caso.

Costo partizione di un vettore

- Il costo $C(i, j)$ di un sottovettore $V[i \dots j]$ di V è pari alla somma dei suoi elementi
  - Esempio: $V = [1, 2, 4, 8, 16, 2]$, $C(1, 3) = 2 + 4 + 8 = 14$.
- Una k-partizione di V è una divisione di V in k sottovettori contigui e non vuoti che coprono totalmente il vettore e non si sovrappongono.
  - Esempio: $V = [2, 3, 7, -7, 15, 2]$, una delle $n - 1$ 2-partizioni è $[2, 3, 7], [-7, 15, 2]$.
- Il costo della k-partizione è il costo massimo dei suoi sottovettori.
  - Nella 2-partizione $[2, 3, 7], [-7, 15, 2]$, il costo dei sottovettori è $2 + 3 + 7 = 12$,
    $-7 + 15 + 2 = 10$,
    quindi il costo della 2-partizione è pari a 12.

Costo partizione di un vettore (Versione formale)

- Il costo $C(i,j) = \sum_{t=i}^{j} V[t]$
- k-partizione di V è una divisione di V in k sottovettori contigui $V[0 \dots j_1], V[j_1+1 \dots j_2], \dots, V[j_{k-1}+1 \dots j_k]$ con $j_0 = 0, j_k = n-1$ e $j_t < j_{t+1}, \forall 1 \leq t < k$.
- Il costo della k-partizione è pari a:
$$\max\{C(j_{t-1}+1, j_t) : \forall 1 \leq t \leq k\}$$

Costo partizione di un vettore

Scrivere un algoritmo che prenda in input un vettore V contenente n interi e un intero k tale che $2 \leq k \leq n$ e restituisca il costo della k-partizione di V di costo minimo.

Esempio: $V = [2, 3, 7, -7, 15, 2]$, $k = 3$
$[2, 3, 7], [-7, 15], [2]$ costo 2+3+7=12
$[2, 3], [7], [-7, 15, 2]$ costo -7+15+2=10 (minima)

Strategia:

1. Soluzione per $k = 2$ (facile: $O(n^2)$, meglio in $O(n)$).
2. Soluzione per $k = 3$ (facile: $O(n^3)$, meglio in $O(n^2)$).
3. Soluzione generale (facile: $O(n^k)$, meglio in $O(kn^2)$).

Vettori ordinati

Si scriva un algoritmo che preso in input n e k, restituisca il numero totale di vettori distinti di lunghezza n, contenenti valori interi compresi fra 1 e k, ordinati dalla relazione $\leq$. Si discuta la correttezza e la complessità dell'algoritmo proposto.
Ad esempio, dati $n = 4$ e $k = 3$, questi sono i possibili vettori ordinati:

$$[1,1,1,1],[1,1,1,2],[1,1,1,3],[1,1,2,2],[1,1,2,3],$$
$$[1,1,3,3],[1,2,2,2],[1,2,2,3],[1,2,3,3],[1,3,3,3],$$
$$[2,2,2,2],[2,2,2,3],[2,2,3,3],[2,3,3,3],[3,3,3,3]$$

e quindi il valore da restituire è 15.