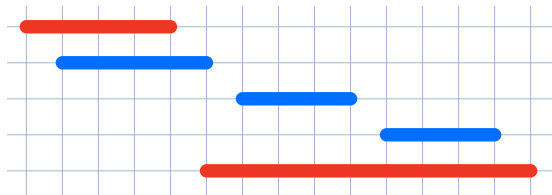


Massima copertura

Si considerino n segmenti sulla retta delle ascisse, dove l' i -esimo segmento inizia nella coordinata $a[i]$ (inclusa) e termina nella coordinata $b[i]$ (esclusa).

Scrivere un algoritmo che prenda in input i vettori a, b e la dimensione n , e restituisca il sottoinsieme di segmenti indipendenti (che non si intersecano) di copertura massimale, ovvero che copre la porzione più grande della retta delle ascisse.

Valutare il costo computazionale dell'algoritmo proposto.



Palindroma

Una stringa si dice palindroma se è uguale alla sua trasposta, cioè se è identica se letta da sinistra e destra o da destra a sinistra.

Scrivere un algoritmo `minpal(ITEM[] s)` che ritorna il numero minimo di caratteri da **inserire** in `s` per rendere `s` palindroma.

Per esempio, input: “casacca”:

- $n = 7$ caratteri: “casaccaACCASAC”
- $n = 6$ caratteri: “casaccaCCASAC”
- $n = 3$ caratteri: “casaccaSAC”
- $n = 2$ caratteri: “ACcasacca”

Notate che non necessariamente i caratteri si inseriscono in testa o in fondo; per esempio, “anta” \rightarrow “antNa”.

Siano dati n dadi, con il dado i -esimo dotato di $F[i]$ facce numerate da 1 a $F[i]$. Scrivere un algoritmo che restituisca il **numero di modi diversi** con cui è possibile ottenere una certa somma X sommando i valori di tutti i dadi.

- Ad esempio, avendo due dadi a quattro facce numerati da 1 a 4, il valore 7 è ottenibile in un solo modo non contando le possibili permutazioni: $3 + 4$.
- Avendo tre dadi, i primi due a 4 e l'ultimo a 6 facce, il valore 8 è ottenibile in cinque modi diversi non contando le possibili permutazioni: $1 + 1 + 6$, $1 + 2 + 5$, $1 + 3 + 4$, $2 + 2 + 4$, $2 + 3 + 3$.

Permutazioni: contatele oppure no, ma siatene consapevoli...