

## Suma Selectiva

14. Dado un conjunto  $X$  con  $|X| = n$  y un entero  $k \leq n$  queremos encontrar el máximo valor que pueden sumar los elementos de un subconjunto  $S$  de  $X$  de tamaño  $k$ . Más formalmente, queremos calcular  $\max_{S \subseteq X, |S|=k} \sum_{s \in S} s$ .
- Proponer un algoritmo *greedy* que resuelva el problema, demostrando su correctitud. Extender el algoritmo para que también devuelva uno de los subconjuntos  $S$  que maximiza la suma.
  - Dar una implementación del algoritmo del inciso a) con complejidad temporal  $O(n \log n)$ .
  - Dar una implementación del algoritmo del inciso a) con complejidad temporal  $O(n \log k)$ .

¿Cómo construimos una solución? Tomo un elemento de  $S$  y lo sumo

Estrategia greedy: Voy agarrando el elemento  $\oplus$  grande de  $X$  que no tomé hasta ahora y lo sumo. Por cada suma me lo guardo en un vector

- ②
- Ordeno  $X$  de forma decreciente
  - Voy recorriendo  $X$  de la 1<sup>ra</sup> pos hasta la  $k$ -ésima  
A medida que recorro sumo el valor y lo guardo en una estructura

Veamos que el algoritmo devuelve una respuesta correcta

Sea  $v^*$  la respuesta que devuelve mi algoritmo QVQ  $v^*$  es correcta

Sup que  $v^*$  No es correcta  $\rightarrow \exists S$  solución /  $\sum_{s \in S} s > \sum_{v \in v^*} v$   
 $|S|=k$   $|v^*|=k$   
 $\rightarrow \exists s \in X / s > v \quad \forall v \in v^*$

$\rightarrow$  ¡ABS! En cada paso voy tomando el valor máximo de  $X$

$\rightarrow$  Mi algoritmo devuelve una respuesta correcta.

⑥

SS(vector X, k)

sort(X);  $\rightarrow O(|X| \log |X|)$   $\rightarrow$  ordena de crecientemente

int suma = 0;

vector res de tamaño k;  $\rightarrow$  complejidad espacial  $O(k)$

for (int j = 0; j < k; j++) {

    suma += X[j];  $O(1)$

    res[j] = X[j];

return suma, res

Complejidad temporal  $O(n \log n)$   
Complejidad espacial  $O(k)$

⑦

SS(vector X, k)

belimHashrand: select k-th  $\rightarrow$  con algo de mediana de las medianas elige el k-ésimo elemento  $\oplus$  grande

vector res (tamaño k)

for (int i = 0; i < |X|; i++) {

    if X[i] > k then

        res.push\_back(X[i])

    endif

endfor

return res

} busco los otros elementos mayores a k