# Algoritmo maximizar número de contenedores cargados (primer algoritmo, el que sí que es óptimo)

## Descripción

* Candidatos: contenedores no escogidos todavía
* Seleccionados: contenedores escogidos hasta el momento
* Función objetivo: maximizar el número de contenedores (maximizar cardinalidad del conjunto de seleccionados)
* Función selección: escoger, de entre los candidatos, el de menor peso, pi
* Función de factibilidad: al incluir el contendor ci de peso pi, la suma de los pesos de los contenedores seleccionados no sea mayor que la capacidad del barco
* Función solución: no se puede incluir ningún contenedor de los candidatos sin que se exceda la capacidad del barco

## Demostración de Optimalidad

Sea S = {c1, c2,…, cn} el conjunto formado por los contenedores de la solución greedy, ordenados en orden de pesos no decrecientes.

Supongamos que S no es una solución óptima al problema.

Existirán, por tanto, una o más soluciones óptimas al problema, diferentes de S.

Llamemos S’ a la solución óptima que contenga el mayor número de elementos coincidentes con S y que estén en la misma posición (en este caso hasta el elemento k).

S = {c1, c2,… ck , ck+1,…, cn} S’ = {c1, c2,… ck , c’k+1,…, c’m}

De forma que hasta k, los contenedores de S’ coinciden con los de S y ambos son escogidos en el mismo orden.

Debido a cómo el algoritmo greedy escoge los candidatos (de los que quedan, siempre el de menor peso) sabemos que entre dos contenedores hasta el número k (tanto en S como en S’) no hay otro contenedor de peso intermedio, pues habría sido escogido antes que el segundo contenedor.

Por todo lo anterior, sabemos que ck+1 c’k+1 y que pk+1 p’k+1 (por cómo escoge la solución greedy los elementos, siempre el de menor peso).

Por tanto, podemos sustituir en S’ a c’k+1 por ck+1, en el caso de que ese contenedor no estuviera en S’,y seguiría siendo S’ una solución válida (no se excedería la capacidad).

En el caso de que ck+1 ya perteneciera a S’, en vez de sustituir los contenedores cambiamos la posición de los contenedores ck+1 y c’k+1 en S’ (cambia el orden en el que se escogen).

En ambos casos S’ pasaría a ser el siguiente conjunto: S’’ = {c1, c2,… ck , ck+1,…, c’m}.

Pero esto es una contradicción, porque dijimos que S’ era la solución óptima que más elementos consecutivos y coincidentes tenía con S y ahora resulta que S’’ tiene un elemento más (ck+1).

**Debido a esto, la hipótesis de partida (que S no es una solución óptima al problema) es falsa y S, la solución greedy, es siempre una solución óptima al problema.**