Analisis y Diseño de Algoritmos

Juan Gutiérrez

November 2, 2022

Algoritmos voraces (Greedy)

Intervalos disjuntos

Problema Max-Intervalos-Disjuntos. Dada una secuencia de intervalos cerrados en la recta, encontrar un subconjunto de intervalos compatibles dos a dos de tamaño máximo.

Intervalos disjuntos

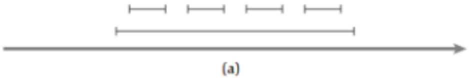

Figure 1: Tomada del libro Kleinberg, Algorithm Design

Intervalos disjuntos

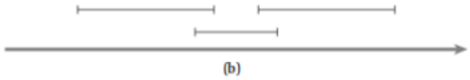

Figure 2: Tomada del libro Kleinberg, Algorithm Design

Intervalos disjuntos

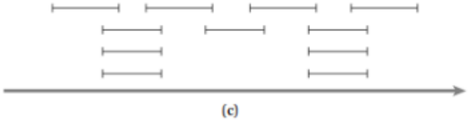

Figure 3: Tomada del libro Kleinberg, Algorithm Design

Intervalos disjuntos

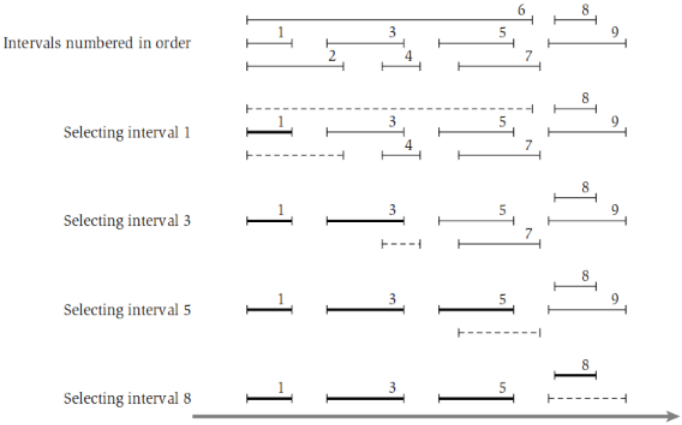

Figure 4: Tomada del libro Kleinberg, Algorithm Design

Elementos de la estrategia Voraz

1. *Elección voraz:* debemos demostrar que siempre existe una solución óptima que contiene a la eleccción voraz
2. *Subestructura óptima*: debemos demostrar que la subsolución dejada es óptima para el subproblema dejado por la elección voraz

Elementos de la estrategia Voraz

Problema Max-Intervalos-Disjuntos. Dada una secuencia de intervalos cerrados en la recta, encontrar un subconjunto de intervalos compatibles dos a dos.

Elementos de la estrategia Voraz

Recibe: un conjunto $\mathcal{I} = \{[s_1, f_1], [s_2, f_2], \ldots, [s_n, f_n]\}$ de intervalos, ordenados de manera creciente por punta final

Devuelve: un subconjunto de intervalos compatibles dos a dos

```
MAX-INTERVALOS-DISJ-REC(I)
if I = ∅
    return ∅
I' = I \ {[s_i, f_i] : s_i ≤ f_1}
return {[s_1, f_1]} ∪ MAX-INTERVALOS-DISJ-REC(I')
```

Elementos de la estrategia Voraz

Lema 3.1 (Elección voraz). *Existe una solución óptima para el problema que contiene el intervalo* $[s_1, f_1]$.

Elementos de la estrategia Voraz

Lema 3.2 (Subestructura óptima). *Si X es una solución óptima al problema que contiene a $[s_1, f_1]$ entonces $X \setminus \{[s_1, f_1]\}$ es una solución óptima al subproblema dejado por la elección voraz.*

Mochila fraccionaria

Problema Mochila-entera. Dado un conjunto $\{1, 2, \ldots, n\}$ de items cada uno con un peso natural w_i, un valor natural v_i y un número natural W, encontrar un subconjunto de items cuya suma de valores es la mayor posible, pero menor o igual a W.

Mochila fraccionaria

Problema Mochila-fraccionaria. Dado un conjunto $\{1, 2, \ldots, n\}$ de items cada uno con un peso natural w_i, un valor natural v_i y un número natural W, encontrar un vector de racionales entre 0 y 1 $(x_1, x_2, \ldots, x_n)$ que maximize $\sum_{i=1}^{n} x_i v_i$ sobre la restricción $\sum_{i=1}^{n} x_i w_i \leq W$

Mochila fraccionaria

Recibe: Una instancia v, w, W del problema MOCHILA-FRACCIONARIA
Devuelve: Una solución óptima para dicha instancia

MOCHILAFRACCIONARIA-GREEDY(v, w, W)

```
for j = n to 1
    if w[j] ≤ W
        x_j = 1
        W = W − w[j]
    else
        x_j = W/w[j]
```

Mochila fraccionaria

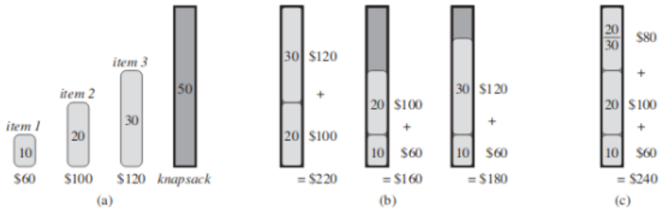

Figure 8: Tomada del libro Cormen, Introduction to Algorithms

Mochila fraccionaria

Lema 4.1 (Elección voraz). *Existe una solución óptima* $(x_1, x_2, \ldots, x_n)$ *al problema tal que* $x_n = \min\{1, W/w_n\}$

Mochila fraccionaria

Lema 4.2 (Subestructura óptima)**.** *Si* $(x_1, x_2, \ldots, x_n)$ *es una solución óptima al problema con* $x_n = \min\{1, W/w_i\}$*, entonces* $(x_1, x_2, \ldots, x_{n-1})$ *es una solución óptima al subproblema dejado con* $W = W - x_n w_n$*.*

Gracias