Ejercicios en clase: Algoritmos voraces (Greedy)

Análisis y Diseño de Algoritmos

22 de junio de 2023

Para cada uno de los ejercicios 1–5 debe

- Indicar su elección voraz
- Diseñar un algoritmo recursivo de acuerdo a su elección voraz
- Enunciar y probar la propiedad de elección voraz
- Enunciar y probar la propiedad de subestructura óptima

Ejercicio 1. Describa un algoritmo eficiente que, dado un conjunto $\{a_1, a_2, \ldots, a_n\}$ de puntos en la recta, determine un conjunto mínimo de intervalos de tamaño 1 que contiene a todos los puntos. Justifique que su algoritmo es correcto usando la propiedades de elección voraz y subestructura óptima. Puede suponer que $a_1 \leq a_2 \leq \cdots \leq a_n$.

Ejercicio 2. Dadas dos secuencias A y B, cada una de las cuales consiste en n enteros positivos, un *cruce* entre A y B es un conjunto de pares ordenados $\{(a_i, b_j) : a_i \in A, b_j \in B\}$, tales que todo elemento en A aparece exactamente una vez, y todo elemento en B aparece exactamente una vez. La *ganancia* de un cruce X es $\prod_{(a_i,b_j)\in X} a_i^{b_j}$. Diseñe un algoritmo voraz que maximiza la ganancia de un cruce. Analize su algoritmo, justificando que es correcto usando las propiedades de elección voraz y subestructura óptima.

Ejercicio 3. Quiero dirigir un carro de una ciudad a otra a lo largo de una carretera. El tanque de combustible del carro tiene capacidad suficiente para cubrir c kilómetros, El mapa de la carretera indica la localización de los puestos de combustible. Queremos encontrar un algoritmo que garantize el viaje con el menor número de abastecimientos.

Más formalmente, usted recibe un arreglo $A[0..n]$ de números reales. El primer punto es el punto de partida, que además es un punto de recarga. Cada uno de los siguientes $n-1$ números indica los puntos de posibilidad de recarga y el último número indica el lugar de destino. Además recibe un número c. Usted empieza en el punto con coordenada $A[0]$. Debe encontrar un arreglo ordenado de índices $B[0..k]$ con el menor tamaño posible, que cumpla que nunca se le va a agotar la gasolina, es decir, $B[0] = 0$ y $A[B[i+1]] \leq A[B[i]] + c$ para $0 \leq i < k$.

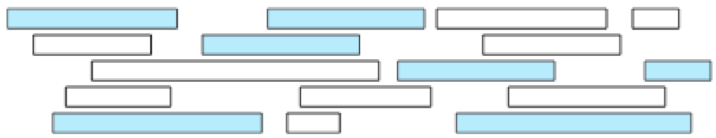
Diseñe un algoritmo voraz. Analize su algoritmo, justificando que es correcto usando las propiedades de elección voraz y subestructura óptima.

Ejercicio 4. Dado un arreglo A de n números naturales, queremos encontrar un arreglo B de tamaño n, que tenga a los elementos de A permutados y que minimize la suma $\sum_{i=1}^{n} iB[i]$.

Diseñe un algoritmo voraz. Analize su algoritmo, justificando que es correcto usando las propiedades de elección voraz y subestructura óptima.

Ejercicio 5. Sea $\mathcal{I}$ un conjunto de n intervalos en la recta real. Decimos que un subconjunto $X \subseteq \mathcal{I}$ de intervalos *cubre* $\mathcal{I}$ si la unión de todos los intervalos en X es igual a la unión de todos los intervalos en $\mathcal{I}$.

Queremos diseñar un algoritmo voraz para encontrar el conjunto más pequeño de intervalos que *cubre* $\mathcal{I}$. Por ejemplo, en la siguiente figura, una solución (no necesariamente óptima) son los 7 intervalos pintados en celeste.



Deberá utilizar la siguiente notación.

Para los intervalos: $\mathcal{I} = \{[s_1, t_1], [s_2, t_2], \ldots, [s_n, t_n]\}$, donde $[s_i, t_i]$ es un intervalo con punta inicial s_i y punta final t_i. Para soluciones devueltas por el algoritmo o utilizadas en las demostraciones: variables X, Y, Y', X'. Pista: ordene previamente sus intervalos según algún criterio (o considere que recibe los intervalos ordenados según ese criterio).

Ejercicio 6. Sea A un conjunto de n pares ordenados en el eje X. Sea B un conjunto de n pares ordenados en el eje Y. Un *matching* para A y B es un conjunto de asociaciones de los elementos de A con los de B, de manera que un mismo par ordenado no puede estar en dos asociaciones. El costo de un matching es la suma de las distancias entre los puntos asociados. Por ejemplo, si $A = \{(0, 4), (0, 3)\}$ y $B = \{(3, 0), (4, 0)\}$, el matching $\{\{(0, 4), (3, 0)\}, \{(0, 3), (4, 0)\}\}$ tiene costo 5+5=10.

Queremos diseñar un algoritmo voraz para encontrar un matching de costo mínimo.

Deberá utilizar la siguiente notación.

Para los conjuntos: $A = \{(0, a_1), (0, a_2), \ldots, (0, a_n)\}$, $B = \{(b_1, 0), (b_2, 0), \ldots, (b_n, 0)\}$. Para soluciones devueltas por el algoritmo o utilizadas en las demostraciones: variables X, X'.

Pista: ordene previamente su entrada según algún criterio (o considere que recibe la entrada ordenada según ese criterio).

Ejercicio 7. Diseñe una **versión simplificada** para resolver el problema de mochila fraccionaria que solo devuelva el valor de la solución encontrada.

Ejercicio 8. Respecto al problema de la mochila.

(a) Indique una instancia del problema de la mochila fraccionária, con al menos tres ítems, en donde la solución brindada por el algoritmo voraz no es una solución óptima para el caso entero.

(b) Para una instancia cualquiera del problema de mochila. ¿Existe alguna relación de desigualdad (mayor, menor, mayor o igual, menor o igual) entre los valores de las soluciones óptimas en los casos enteros y fraccionarios? Justifique.