Ejercicios en clase: Algoritmos voraces (Greedy)

Análisis y Diseño de Algoritmos

29 de septiembre de 2020

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.

Solución.

1. Elección voraz: Seleccionar $[a_1, a_1 + 1]$.
2. Algoritmo recursivo.
   Recibe: Un conjunto $A = \{a_1, \ldots, a_n\}$ de puntos en la recta real tal que $a_1 \leq \cdots \leq a_n$
   Devuelve: Un conjunto de intervalos unitarios de tamaño mínimo que cubre A
   VORAZ-SEGMENTOS(A)
   ```
   if A = ∅
       return ∅
   A' = {a_i ∈ A : a_i > a_1 + 1}
   return {[a_1, a_1 + 1]} ∪ VORAZ-SEGMENTOS(A')
   ```
3. Prueba de la elección voraz.

   Lema 0.1 (Elección voraz)**.** *Existe una solución óptima que contiene a* $[a_1, a_1 + 1]$.

   Demostración. Sea X una solución óptima. Si $[a_1, a_1 + 1] \in X$ entonces no hay más que mostrar. Suponga entonces que $[a_1, a_1 + 1] \notin X$. Como X es una solucion, existe un intervalo $[p, p + 1] \in X$ que contiene a a_1, osea $p \leq a_1 \leq p + 1$. Mostraremos que $X' = X \setminus \{[p, p + 1]\} \cup \{[a_1, a_1 + 1]\}$ es también una solución al problema.

   Para esto, debemos probar que todo $a_i \in A$ está en algún intervalo de X'. Sea $a_i \in A$. Si $a_i \notin [p, p+1]$ entonces está en algún intervalo de $X \setminus \{[p, p+1]\}$. Dicho intervalo también está en X' y por lo tanto a_i está cubierto por algún intervalo de X'. Si $a_i \in [p, p + 1]$ entonces

   $$a_1 \leq a_i \leq p + 1 \leq a_1 + 1,$$

   y portanto $a_i \in [a_1, a_1 + 1]$. Luego X' es una solución y como $|X| = |X'|$, entonces X' es una solución óptima. □
4. Prueba de subestructura óptima

Lema 0.2 (Subestructura óptima)**.** *Si X es una solución óptima para A que contiene a $\{[a_1, a_1+1]\}$, entonces $X \setminus \{[a_1, a_1+1]\}$ es una solución óptima para A'.*

Demostración. Suponga por contradicción que $X' = X \setminus \{[a_1, a_1+1]\}$ no es óptima en A'. Entonces existe una solución Y' para A' tal que $|Y'| < |X'|$. Entonces $Y = Y' \cup \{[a_1, a_1+1]\}$ es una solución óptima para A. Pero $|Y| = |Y'| + 1 < |X'| + 1 = |X|$, una contradicción. □

Ejercicio 2. Dados dos conjuntos A y B, cada uno de los cuales tiene n enteros positivos, un *cruce* entre A y B es un conjunto de pares ordenados $\{(a_i, b_j) : a_i \in A, b_j \in B\}$, tal 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.

1. Elección voraz: elegir (a_{i*}, b_{j*}) tal que a_{i*}, b_{j*} son elementos máximos en A y B respectivamente.

2. Algoritmo voraz.

   Recibe: Dos conjuntos de números $A = \{a_1, \ldots, a_n\}$ y $B = \{b_1, \ldots, b_n\}$.
   Devuelve: Un cruce entre A y B con ganancia máxima.

   ```
   VORAZ-CRUCE(A, B)
   if A = ∅
       return ∅
   Sea a_{i*} en elemento máximo en A
   Sea b_{j*} en elemento máximo en B
   A' = A \ {a_{i*}}
   B' = B \ {b_{i*}}
   return {(a_{i*}, b_{j*})} ∪ VORAZ-CRUCE(A', B')
   ```

3. Prueba de la elección voraz.

   Lema 0.3 (Elección voraz)**.** *Existe una solución óptima que contiene a (a_{i*}, b_{j*}), donde a_{i*} es un elemento máximo en A y b_{j*} es un elemento máximo en B*

   Demostración. Sea X una solución óptima al problema. Si $(a_{i*}, b_{j*}) \in X$ entonces no hay nada que probar. Suponga entonces que $(a_{i*}, b_{j*}) \notin X$. Como X es una solución, entonces existen elementos $a_i \in A, b_j \in B$ tales que $(a_{i*}, b_j), (a_i, b_{j*}) \in X$.

   Sea $X' = X \setminus \{(a_{i*}, b_j), (a_i, b_{j*})\} \cup \{(a_{i*}, b_{j*}), (a_i, b_j)\}$.

   Es claro que X' es también una solución para el problema. A continuación mostraremos que su ganancia es tanto como la ganancia de X. Note que

   $$\frac{ganancia(X')}{ganancia(X)} = \frac{a_{i^*}^{b_{j^*}}}{a_{i^*}^{b_j}} \cdot \frac{a_i^{b_j}}{a_i^{b_{j^*}}} = (a_{i^*}/a_i)^{b_{j^*}-b_j} \geq 1.$$

Luego, X' es una solución óptima que contiene a (a_{i*}, b_{j*}), lo que queríamos demostrar. $\square$

4. Prueba de subestructura óptima

Lema 0.4 (Subestructura óptima)**.** *Si X es una solución óptima para (A, B), entonces $X \setminus \{(a_{i*}, b_{j*})\}$ es una solución óptima para (A', B').*

Demostración. Suponga por contradicción que $X' = X \setminus \{(a_{i*}, b_{j*})\}$ no es óptima en (A', B'). Entonces existe una solución Y' para (A', B') tal que $ganancia(Y') > ganancia(X')$. Entonces $Y = Y' \cup \{(a_{i*}, b_{j*})\}$ es una solución para (A, B). Pero

$$ganancia(Y) = ganancia(Y') \cdot a_{i^*}^{b_{j^*}} > ganancia(X') \cdot a_{i*}^{b_{j*}} = ganancia(X),$$

una contradicción. $\square$