miércoles, 13 de noviembre de 2024 09:02

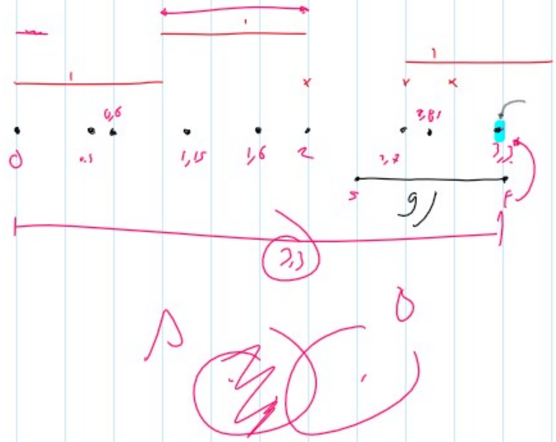
Ejercicio 1. Describa un algoritmo eficiente que, dado un conjunto $\{a_1, a_2, \ldots, a_n\}$ de puntos en la recta, determine un conjunto mínimo de intervalos de tamaño 1 que contiene a todos los puntos. Justifique que su algoritmo es correcto usando la propiedades de elección voraz y subestructura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

Q_1: $A = \{a_1, \ldots, a_n\}$ set puntos. // $a_1 \le \cdots \le a_n$

Q_2: $\mathcal{I} = \{\text{intervalos}\}$ Tq $|I|=1 \wedge A \subseteq \bigcup_{I\in\mathcal{I}} I$

Q_3: $|\mathcal{I}|$ mínima

Q_4 $g = [a_n - 1, a_n] \leftarrow$



Q_5: $A' = A - g$

$= A - [a_n - 1, a_n]$

$= A - \{a_i \in A \mid a_i \in [a_n - 1, a_n]\}$

$= A - \{a_i \in A \mid a_n - 1 \le a_i \le a_n\}$

$= A - \{a_i \in A \mid a_i \ge a_n - 1\}$

$= \{a_i \in A : a_i < a_n - 1\}$

= Todos los puntos de A que no pertenezca al intervalo de g.c.

Algo (A)

```
if (|A| == 0) return ∅
else
    g = [a_n - 1, a_n]
    A' = {}
    for i=1 To n
        if (a_i < a_n - 1) A'.pb(a_i)

    return {g} ∪ Algo(A')
```

G. C. P.

$\exists$ O opT sol' Tq contenga al greedy

$\exists$ O opT sol Tq $g \in O$.

Prueba

. Sea X una sol opT cualquiera. Hay 2 casos

_ Caso 1: $g \in X$. Facilito $O = X$

_ Caso 2: $g \notin X$

[s, f)

- Caso 1: $g \in X$. Facilito $\cup = X$
- Caso 2: $g \notin X$

Como X es sol del problema X contiene un intervalo J que cubre a a_n.

Además, como X no contiene al greedy choice, los extremos de J están ubicados a la derecha de los de g.

Construimos X' como el resultado de intercambiar los objetos J y g en X.

- Claramente g es elemento de X'
- X' es un conjunto de intervalos unitarios
- Al retirar J de X, hemos descubierto algunos puntos de A, pero por lo visto arriba, como g es un desplazamiento más a la izquierda de J, entonces todos los puntos ~~es posiblemente más~~ seguirían siendo cubiertos por g. Con esto concluimos que X' sigue cubriendo a A.

$\Rightarrow \exists J \in X$ Tq $a_n \in J = [s,f] \rightarrow s \le a_n \le f \wedge f-s=1$

También tenemos: $J \neq g$

$\Rightarrow \boxed{s > a_{n-1} \wedge f > a_n}$

Sea $X' = X - \{J\} \cup \{g\}$

$\Rightarrow$

- $g \in X'$
- $I \in X' \rightarrow |I| = 1$
- Sea $p \in A \cap J$

$\Rightarrow s \le p \le f$

$p \in A \longrightarrow p \le a_n$ ✓

$p \ge s = f-1 > a_n - 1 \rightarrow p > a_n - 1$

$\Rightarrow a_n - 1 \le p \le a_n \Rightarrow p \in A \cap g.$

$\Rightarrow A \subseteq \bigcup_{I \in X} I$ ✓

$a_n - 1 \le p \le a_n$

quiero $p \in g$.

O.S.P

- X sol opt $P(A)$ que contiene a g
- $X' = X$ sin g . A' subproblema

$\Rightarrow X'$ sol opt $P(A')$

X sol op $P(A)$ Tq $g \in X$

. $X' = X - \{g\} \wedge A' = \underline{A - \{a_i : a_i \in g\}}$

$\Rightarrow X'$ sol opt $P(A')$

Prueba:

- Supongo x contradicción que X' no es opt. de $P(A')$
- $\exists Y'$ sí es opt. $P(A')$ $\leftarrow |Y'| < |X'|$
- Sea $Y = Y' \cup \{g\}$.
- Como Y' cubre A' y g cubre $g \Rightarrow Y$ cubre $A' \cup g$ que a su vez cubre A.

$\Rightarrow Y$ es una sol.

Además: $|Y| = |Y'| + 1 < |X'| + 1 = |X|$

Esto implica que X nunca fue opt!! T.T. $(\Rightarrow \Leftarrow)$.

Además: $|Y| = |Y'| + 1 < |X'| + 1 = |X|$

Esto implica que X nunca fue opt!! T.T. $(\Rightarrow)$ $(\Leftarrow)$.

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.

Q_1: A, B : $|A| = n = |B|$ de enteros positivos.

Q_2: Un cruce $X = \{(a_i, b_j) : a_i \in A, b_j \in B\}$

Q_3: Max $gan(X) = \prod_{(a_i,b_j)\in X} a_i^{b_j}$

$A = \{1, 3, 4, 6, 7\}$

$B = \{3, 3, 5, 6, 10\}$

$6^{10} \times 7^{6} = 6^{6} \times 7^{10}$

One (a_n, b_n), $a_1 \le \dots \le a_n$ $\wedge$ $b_1 \le \dots \le b_n$

Q_{n-1}: $A[1:n-1]$, $B[1:n-1]$

G. C. P

$\exists X$ sol opt tq $(a_n, b_n) \in X$

Sea X un opt arbitrario

Caso 1: $(a_n, b_n) \in X$ ✓

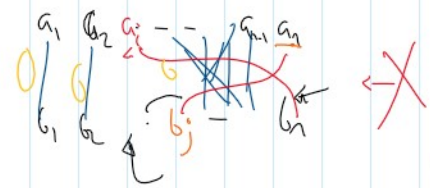
Caso 2: $\boxed{(a_n, b_n) \notin X}$

Como X es un cruce, y a_n y b_n no están emparejados

$\Rightarrow \exists\, i, j < n$ tq $(a_n, b_j), (a_i, b_n) \in X$

$X' = X - \{(a_n, b_j), (a_i, b_n)\} \cup \{(a_n, b_n), (a_i, b_j)\}$

- $g \in X'$ ✓
- X' es un cruce ✓
- ¿$gan(X') \ge gan(X)$? $\longleftrightarrow$ ¿$\frac{gan(X')}{gan(X)} \ge 1$?



$g_{an}(X)$

$$g_{an}(X') = g_{an}(X) \cdot \frac{1}{a_n^{b_j}} \cdot \frac{1}{a_i^{b_n}} \times a_n^{b_n} \cdot a_i^{b_j}$$

up

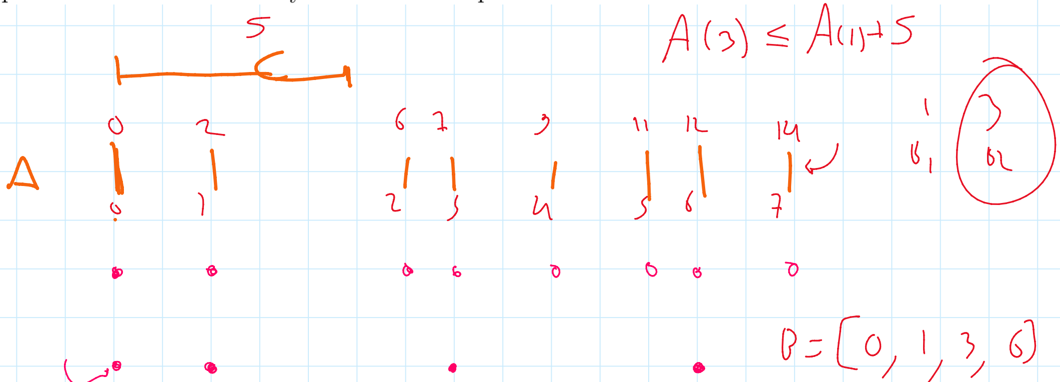
$$= g_{an}(X) \cdot \frac{a_n^{b_n - b_j}}{a_i^{b_n - b_j}} = \left(\frac{a_n}{a_i}\right)^{\overbrace{b_n - b_j}^{\geq 0}} \cdot g_{an}(X) \geq 1 \cdot g_{an}(X).$$

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.

$A(3) \leq A(1) + 5$



$B = [0, 1, 3, 6]$

0 1 2 7