Ejercicios Básicos

Para cada uno de los siguientes problemas:

- Diseñe un enfoque voraz para resolverlo, e impleméntelo en pseudocódigo.
- Demuestre que el algoritmo voraz diseñado encuentra la solución óptima, o muestre un contra-ejemplo en el que no la encuentra.

1. Dada una fracción a/b se desea expresarla como la suma de un número mínimo de fracciones distintas con numerador unidad, e.g., $3/5 = 1/2 + 1/10$.

EJEMPLO

$$\frac{3}{5} = \frac{1}{2} + \frac{1}{10}$$

EL RAZONAMIENTO QUE LLEVARÍA A CABO SERÍA IR COGIENDO EL MENOR K, SIENDO $\frac{1}{k}$ PARA ACERCARNOS AL NUMERO SIN PASARNOS Y SEGUIREMOS SUMANDOLE ALGUNA K SIENDO SUMADO SI TE PASAS (K+1) HASTA QUE DE DE RESULTADO EL NUMERO.

EN MI SOLUCIÓN TENDRE UNA LISTA DE X ELEMENTOS QUE SEAN LAS K (ES DECIR LOS DENOMINADORES) $\{x_0, x_1, x_2, x_3 \dots x_n\}$

$l = \{l_i\}$ (LISTA DE ELEMENTOS K)

Inicial = $\{\}$ (AL PRINCIPIO NO HABRÁ ELEMENTOS EN LA LISTA)

$$\text{Final}(l) = r - \sum_{i=0}^{l.\text{lenght}-1} \frac{1}{l_i} = 0$$

r: NÚMERO AL QUE HAY QUE ALCANZAR

$\sum$: TODA LA LISTA

$$\text{Válida}(l, K) = r - \sum_{i=0}^{l.\text{lenght}-1} \frac{1}{l_i} + \frac{1}{K} \geq 0$$

AÑADIENDO EL SIGUIENTE DEBE DE SER O MAYOR QUE 0 O IGUAL A 0, POR LO QUE SERÍA FINAL.

NO SERA VÁLIDA SI AL SUMARLE EL SIGUIENTE DA UN NÚMERO NEGATIVO

$$\text{Selección}(l) = \underset{K \in \mathbb{N}}{K} \quad \underset{1 \leq k \leq \infty}{\min} \quad \frac{1}{K} \leq r \left(\boxed{< \frac{1}{k-1}} \right)$$

DUDOSA

PUESTO POR EL PROFESOR

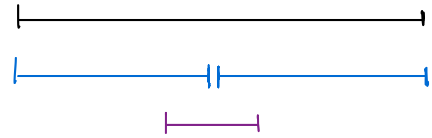
$\vee$ VÁLIDA (L, K)

AÑADIRÍA YO.

2. Durante la preparación de la semana cultural de Informática se recibe un conjunto de solicitudes para usar el salón de actos. Cada una de estas solicitudes requiere uso exclusivo del salón, y está determinada por una hora de comienzo c_i y una hora de finalización f_i. Determinar el mayor número de solicitudes que se pueden satisfacer.

TENGO $S = \{[c_i, f_i]\}$ TEMOS DISTINTO HORARIO Y TENDREMOS QUE COGER EL QUE ACABE ANTES

SE PODRÍA PENSAR QUE HABRÍA QUE COGER LA QUE MENOS DURE PERO AQUÍ HAY UN CONTRAEJEMPLO



SERIA MÁS RENTABLE COGER LAS DOS GRANDES ANTES QUE AL PEQUEÑO YA QUE EN EL AZUL HAY DOS ACTIVIDADES Y EN EL MORADO SOLO UNA

POR LO TANTO UNA POSIBLE SOLUCIÓN:

$L = \{l_i\}$ $l_i = j \equiv$ LA TAREA iesima es S_j - $S_j = \{[c_j, f_j]\}$

Lista compuesta con las actividades.

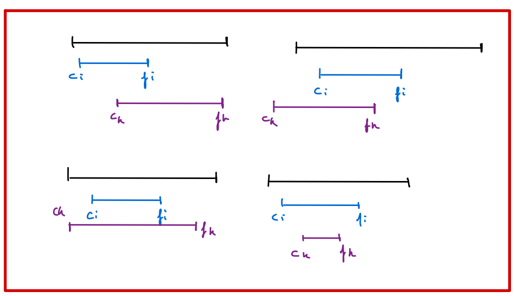
Inicial $= \{\ \}$ (AL PRINCIPIO NO HABRÁ ELEMENTOS EN LA LISTA)

Final = SERA FINAL CUANDO EL CONJUNTO DE TAREAS POR SELECCIONAR SEA VACIO

CONJUNTO DE ACTIVIDADES

TODO NEGADO YA QUE SON LOS CASOS QUE **NO**

$$\text{VÁLIDA}(L,k) = \forall l_i \in L : \overline{(f_i > c_k \wedge c_i < c_k) \vee (f_k > c_i \wedge f_k < f_i) \vee (c_i > c_k \wedge f_i > c_k \wedge c_i < f_k) \vee (c_k > c_i \wedge f_k < f_i)}$$



CASOS QUE NO PUEDEN OCURRIR PARA QUE SEA VÁLIDA. IGUAL HAY ALGUNA REDUNDANTE.

$$\text{SELECCIÓN}(L) = k \,/\, S_k \in S \wedge \text{VÁLIDA}(L,k) \wedge$$
$$\wedge \nexists\, S_i \in S \,/\, S_i \notin L \wedge f_i < f_k$$

BASICAMENTE ELEGIR UNA QUE SEA VÁLIDA QUE PERTENEZCA A S (LAS ACTIVIDADES) Y NO EXISTE OTRO Si que ACABE ANTES QUE LA QUE HEMOS ELEGIDO.

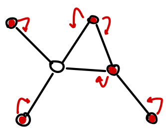
3. Dado un grafo $G(V, E)$ se desea encontrar un conjunto $S \subseteq V$ de tamaño mínimo tal que cada arista $(u, v) \in E$ tiene al menos uno de sus extremos en S.

SIENDO G – GRAFO, QUE SE FORMA CON (V, E), DONDE
V – VERTICES
E – ARISTAS

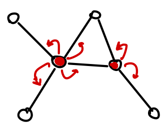
TENEMOS QUE ENCONTRAR UN SUBGRAFO MINIMO.

EL PLANTEAMIENTO DE ESTE EJERCICIO SERIA IR REDUCIENDO EL NÚMERO DE VERTICES QUE TIENEN MAYOR GRADO.

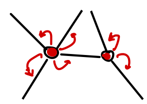
EJEMPLO



ESTA "PODRÍA" SER UNA SOLUCIÓN PERO NO LA OPTIMA



$\Rightarrow$



ESTE SI SERÍA EL OPTIMO YA QUE HEMOS EMPLEADO LOS VERTICES CON MAYOR GRADO

$L = (\{l_i\}, G)$ — LO QUE ME QUEDA DEL GRAFO — $l_i = j \equiv$ el vertice $V_j \in G$

Inicial $= (\{\}, G(V,E))$ — PUEDES PONERLO O NO ("SUPONGO") ES SOLO PARA HACERLO MÁS FÁCIL DE ENTENDER

Final $= ((L, G)) = E = \emptyset$ } CUANDO EL NÚMERO DE ARISTAS POR CUBRIR SEA NULO.

Válida $((L, G), k) = k \notin L \wedge k \in V$ } k ES UN VERTICE A AÑADIR A LA LISTA Y NO ESTA YA EN ELLA

Selección $((L, G)) = (L + \{k\}, G')$: $k \in V \wedge gr(k, G) = \alpha \wedge \nexists u \in V / gr(u, G) > \alpha$ — NO PUEDE TENER MAYOR GRADO

G' — SE IRA REDUCIENDO. $\wedge$ $G' =$ extraer (G, k)

4. Dado un grafo $G(V, E)$, un *coloreado* del mismo consiste en asignar un color a cada vértice de manera que si $(u, v) \in E$ entonces u y v tienen distinto color. Se desea encontrar un coloreado que emplee el mínimo número de colores.

SIENDO G – GRAFO, QUE SE FORMA CON (V, E), DONDE
V – VERTICES
E – ARISTAS

POR LO TANTO MI SOLUCIÓN VA A SALIR EN UNA LISTA DONDE LOS NÚMEROS SON LOS COLORES $\{1, 2, 1, 3\}$
VERTICE 1 DEL COLOR 1

$L = \{l_i\}$ $\quad l_i = j \equiv$ EL VERTICE i ESTA COLOREADO DEL COLOR j

Inicial $= \{\}$ (AL PRINCIPIO NO HABRÁ ELEMENTOS EN LA LISTA)

FINAL $=$ $L.\text{length} == |V|$ CUANDO EL TAMAÑO DE LA LISTA SEA IGUAL AL CARDINAL DE CUANTOS VERTICES HAY.

$$\text{VÁLIDA}(L, c) = \forall u \in V \,/\, \overbrace{(V.\text{length}, u) \in E \vee (u, V.\text{length}) \in E}^{\text{ESTAR RELACIONADO CON EL QUE VAS A COLOREAR}} \wedge \underbrace{u \in L}_{\text{ESTA COLOREADO}} \Rightarrow l_u \,!= c$$

c: COLOR NUEVO A AÑADIR

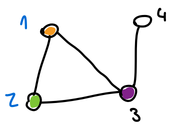
$$\text{SELECCIÓN}(L): \min\left(c \in [1, \max(l) + 1] \,/\, \text{VALIDA}(L, c)\right)$$

DEBE DE SER VÁLIDO ESE COLOR

YA QUE NUESTRO OBJETIVO ES TENER EL MIN $N^{\underline{o}}$ DE COLORES

CLARO YA QUE TENDRÉ QUE COGER EL MINIMO DENTRO DEL RANGO 1 —— HASTA EL COLOR MÁXIMO NUMERICO QUE HAYA, MÁS 1 POR SI ESTA CONECTADO

EJEMPLO



1º NUESTRA LISTA SERIA $\{\overset{1}{1}, \overset{2}{2}, x\}$

ESA x SERA O EL 1 (NARANJA) O HASTA EL MAX DE LA LISTA → 2 (VERDE) + 1 QUE EN ESTE CASO SERIA

$\{1, 2, \overset{2+1}{3} \ldots\}$

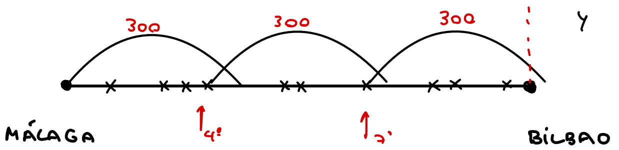
COLOR 3 – EL MORADO

5. Vamos a ir en coche desde Málaga a Bilbao a ver el Guggenheim. Salimos con el depósito lleno, y sabemos que con ese combustible podemos recorrer M kilómetros. Tenemos un mapa en el que figuran los puntos kilométricos $k(1), \cdots, k(n)$ en los que hay gasolineras, y queremos determinar en cuáles de ellas hay que repostar para llegar al destino haciendo el menor número de paradas.

HAY VARIAS CASUISTICAS EN ESTE EJERCICIO

PERO EN ESTE CASO:

EJEMPLO



LAS CRUCES SON LOS PUNTOS (KILÓMETROS DONDE HAY GASOLINERA) Y CUANDO NOS PASEMOS DEL • BILBAO HABRÍA TERMINADO.

SOLUCIÓN LISTA {4, 7} (HEMOS TENIDO QUE PARAR EN LA 4 Y 7.)

TAMBIÉN TENDREMOS UN ARRAY g

20	100	120	230	260	300	360	420
0	1	2	3	$\boxed{4}$	5	6	7

POR LO TANTO

$L = \{ l_i \}$ $\quad l_i = j \equiv$ iésima PARADA EN LA GASOLINERA j

Inicial $= \{\}$ "0" $\underbrace{\{0\}}$

SI TIENES EN CUENTA QUE RELLENAS EN MÁLAGA EN LA GASOLINERA 0.

$$\text{FINAL}(L) = \underbrace{g[g.\text{length} - 1]}_{\text{LA ULTIMA GASOLINERA}} - \underbrace{g[l[l.\text{size} - 1]]}_{\text{LA ULTIMA GASOLINERA DE LA LISTA DONDE HEMOS HECHADO}} \leq M$$

LA DISTANCIA DE AUTONOMIA DEBE DE SER MENOR O M

$$\text{VÁLIDA}(L, K): \quad K \in [1, g.\text{length} - 1] \;\wedge\; \forall j \in L \;/\; K > j$$

YA QUE NO VAS A COGER UNA GASOLINERA ANTERIOR.

$j = L.\text{size} - 1$

$$\text{SELECCIÓN}(L) = \max\, K \in (j < K \leq g.\text{length} - 1) \;/\; g[K] - g[i] \leq M$$

NO SE SI SINTACTICAMENTE ESTA BIEN BIEN ESCRITO PERO ES COGER UN K DENTRO DEL RANGO DEL MAYOR DE LA LISTA Y DEL Nº MAX DE GASOLINERAS

TAL QUE LOS KILOMETROS DE ESE PUNTO (GASOLINERA) MENOS LOS DE LA ANTERIOR DONDE HE REPOSTADO SEAN MENOR QUE M (AUTONOMIA EN KM DEL COCHE)

6. Tenemos que planificar la gira veraniega del grupo de teatro de Informática. Tenemos n propuestas, cada una de las cuales consta de una fecha de inicio d_i, un número de actuaciones a_i (siempre una al día), y una oferta económica m_i. Seleccionar las propuestas que aceptaremos para maximizar el beneficio económico.

NOS DAN N EVENTOS DONDE CADA UNO TIENE SU FECHA INICIO (d_i), EL NÚMERO ACTUACIONES (a_i) y oferta económica (m_i)

TENEMOS QUE MAXIMIZAR LO ECONÓMICO (m_i), POR LO TANTO COGEREMOS LA ACTIVIDAD QUE MÁS OFERTA NOS DE, AUNQUE NO TIENE PORQUE SER LA ÓPTIMA

DATOS QUE ME DAN: n - actuaciones

d_i

a_i

m_i

$L = \{ l_i \}$ $\quad l_i : j$: EL iesimo EQUIVALE A LA ACTIVIDAD j

Inicio: $\{ \}$ (AL PRINCIPIO NO HABRÁ ACTIVIDADES EN LA LISTA)

$\text{FINAL}(L) = d_i[L.\text{length} - 1] == d_n$

SUPONGO QUE SERA FINAL CUANDO LA ULTIMA FECHA DE LA LISTA COINCIDA CON LA MÁXIMA FECHA DEL ARRAY DE FECHAS (YA QUE VAN POR DÍAS)

$\text{VÁLIDA}(L, k) = \underbrace{k \in N}_{\text{ES UNA DE LAS POSIBLES ACTIVIDADES}} \wedge \text{TRUE}$ (CUALQUIER ACTIVIDAD PUEDE SER VALIDA YA EN SELECCIÓN, PONDREMOS LA CONDICIÓN DE PORQUE COGER A UNO O A OTRO)

$\text{SELECCIÓN}(L) : k \in N \wedge \underbrace{\forall i \in L \cdot k / \nexists\, d_i == d_k}_{\text{NO TENGAN LA MISMA FECHA}} \wedge \underbrace{\forall u \in N / d_u == d_k \Rightarrow m_k \geq m_u}_{\text{NO EXISTA PARA ESA FECHA OTRO CON MAYOR OFERTA}}$

7. Definamos un *intervalo unitario* como un subconjunto de la recta real de la forma $[a,b]=\{z \mid a \le z \le b\}$, donde $b-a=1$. Por ejemplo, $[\frac{5}{4},\frac{9}{4}]$ es un intervalo unitario. Considere el *problema del cubrimiento de intervalo* donde la entrada es un conjunto $\{x_1,x_2,...,x_n\}$ de n puntos en un orden arbitrario en la recta real, y la salida es un conjunto formado por el menor número de intervalos unitarios que cubren todos los puntos de entrada (en el sentido de que cada punto está en al menos uno de los intervalos). Proponga un algoritmo voraz para resolver el problema (de forma correcta si es posible)

Como un ejemplo, supongamos que la entrada es $\{\frac{7}{4},\frac{7}{2},\frac{1}{2},2,\frac{3}{2},0\}$. Entonces $\{[0,1],[\frac{5}{4},\frac{9}{4}],[\frac{5}{2},\frac{7}{2}]\}$ es una salida (i.e., respuesta) válida al problema planteado pues se puede comprobar fácilmente que estos tres intervalos cubren todos los seis puntos de entrada, y también que no existen dos intervalos unitarios que cubran todos los puntos de entrada.

OBJETIVO ES ENCONTRAR DENTRO DE LOS PUNTOS N QUE NOS PASAN POR EL ARRAY LOS QUE SU RANGO UNITARIO CUBREN TODOS, EL EJEMPLO PROPORCIONADO NO ESTA BIEN RESUELTO.

BASICAMENTE SE LE SUMARA 1 $\{0\} = [0, 0+1] = [0,1] = 1-0=1$ UNITARIO

POR LO TANTO:

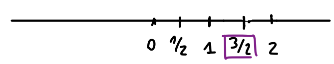
$L = \{l_i\}$ $\quad l_i := j \equiv$ AL inicio del intervalo iesimo sera $j \,/\, [j, j+1] \in S$ Y ES UNITARIO

INICIAL = $\{\}$

INDICE DE COMIENZO

FINAL $(L) = \nexists x_i \in X \,/\, \exists j \in L \quad x_i - j \le 1$

ARRAY INICIAL CON LOS DATOS



PARA $j=1$ $\nexists x_i \in X \,/\, 1 - 3/2 = 1/2 \le 1$

SINO PASASE ESO NO SERÍA FINAL Y HABRÍA QUE AÑADIR OTRO INTERVALO

VÁLIDA $(L,K) =$ TRUE $\wedge$ $K \in X$ (EN ESTE NO TENDRÍA MUCHO SENTIDO YA QUE CUALQUIER NÚMERO PODRÍA SER UNITARIO, TENDREMOS QUE MODIFICAR LA SELECCIÓN)

SELECCIÓN $(L) = \min\left(k \in X \,/\, \nexists l_i \in L \quad x - l_i > 1\right)$

"HAY QUE COGER EL MENOR X QUE NO ESTE CUBIERTO"

8. Se tienen n procesos, cada uno de los cuales tiene duración t_i, que se desean ejecutar en un procesador. Dado un cierto orden de ejecución de los procesos, el tiempo de estancia en el sistema de cada uno de los mismos es el tiempo de espera hasta ser ejecutados más el tiempo de ejecución. Se desea encontrar un orden de ejecución que minimice la suma de los tiempos de estancia en el sistema de todos los procesos.