



1 EVALUACIÓN

Acerca del método de evaluación

Los problemas y requisitos que se detallan a continuación son solamente de carácter opcional. El postulante tendrá la opción de elegir cuales resolver y aún sin ser excluyentes para la contratación, es un plus a la hora de la decisión final.

Se podrán resolver en cualquier lenguaje y plataforma y el entregable deberá ser solo el código y una descripción de las herramientas utilizadas. Ej (se resolvió con C# bajo .NET 4.5)

Happy coding!!!

Original



2 PROBLEMAS

2.1 ALGORITMOS

Clavando tableros

Se le dan dos arreglos A y B sin cero, no vacías, que consisten en N enteros.

Estos arreglos representan N tableros. Más precisamente, A [K] es el comienzo y B [K] el final del K-ésimo tablero.

A continuación, se le da un arreglo no vacío cero-indexado C que consta de M enteros. Este arreglo representa M clavos.

Más precisamente, C [I] es la posición donde se puede martillar en el clavo I-ésimo.

Decimos que un tablero (A [K], B [K]) es clavado si existe un clavo C [I] tal que $A [K] \leq C [I] \leq B [K]$.

El objetivo es encontrar el número mínimo de clavos que deben utilizarse hasta que todos los tableros estén clavados.

En otras palabras, debe encontrar un valor J tal que todos los tableros serán clavados después de usar sólo los primeros clavos.

Más exactamente, para cada tablero (A [K], B [K]) tal que $0 \leq K < N$, debería existir un clavo C [I] tal que $I < J$ y $A [K] \leq C [I] \leq B [K]$.

Por ejemplo, dados los arreglos:

A[0] = 1 B[0] = 4
A[1] = 4 B[1] = 5
A[2] = 5 B[2] = 9
A[3] = 8 B[3] = 10

Se pueden representar 4 tableros: [1, 4], [4, 5], [5, 9] y [8, 10].

Dado el arreglo C:

C[0] = 4
C[1] = 6
C[2] = 7
C[3] = 10
C[4] = 2

Original



Si usamos los siguientes clavos:

0, entonces los tableros [1, 4] y [4, 5] serán clavados.
0, 1, entonces los tableros [1, 4], [4, 5] y [5, 9] serán clavados.
0, 1, 2, entonces los tableros [1, 4], [4, 5] y [5, 9] serán clavados.
0, 1, 2, 3, entonces todos los tableros serán clavados.

Así, cuatro es el número mínimo de clavos que, usados secuencialmente, permiten que todos los tableros sean clavados.

Escribir una función:

int solution (int A [], int B [], int N, int C [], int M);

que, dada dos arreglos A y B no vacías de índice cero que consisten en N enteros y un arreglo C
cero no vacía compuesta por M enteros,
devuelve el número mínimo de clavos que, usados secuencialmente, permiten que todos los
tableros sean clavados.

Si no es posible clavar todos los tableros, la función debe devolver -1.

Por ejemplo, dados los conjuntos A, B, C tales que:

A[0] = 1	B[0] = 4
A[1] = 4	B[1] = 5
A[2] = 5	B[2] = 9
A[3] = 8	B[3] = 10

C[0] = 4
C[1] = 6
C[2] = 7
C[3] = 10
C[4] = 2

La función debe devolver 4, como se explicó anteriormente.

Asumir que:

N y M son números enteros dentro del intervalo [1, 30.000];
Cada elemento de los arreglos A, B, C es un número entero dentro del rango [1, 2 *
M]; $A[K] \leq B[K]$.

Original



Buscando anagramas

Se provee una cadena de caracteres de largo N y una subcadena de largo M . Se desea saber cuántas veces aparece la subcadena o un anagrama de la misma dentro de la cadena principal.

Por ejemplo, si se tiene la cadena A y la subcadena B

$A = \text{"hola, que buena ola Laomir"}$

$B = \text{"OAL"}$

El resultado deberá ser 3, dado lo siguiente

$A = \text{"hola, que buena } \mathbf{ola} \mathbf{Laomir"}$

Escribir una función

`Int solution(string A, string B)`

que devuelva la cantidad de veces que aparece B en A , o un anagrama de B en A .

Asumir:

No hay distinción entre mayúsculas y minúsculas $N > M$.

Original



2.2 DISEÑO

Pasar la siguiente imagen a HTML, utilizando CSS. Se puede utilizar cualquier plugin para animar los componentes de alguna manera (opcional).

Asumir que las imágenes de fondo, billetes y paleta de colores son proveidas.

Resumen del monto recibido

CANTIDAD	VALOR	PESOS	CANTIDAD	VALOR	PESOS
0		0	100		5.000
5		25	200		20.000
10		100	50		10.000
20		400	100		50.000

CANTIDAD TOTAL

328

MONTO TOTAL

\$53.564

Agregar más billetes

Aceptar

Original



2.3 MODELADO

1- Diagrama de secuencia

Considere una transacción de extracción de efectivo mediante un cajero automático. Presentar un diagrama de secuencia que sirva para sistematizar dicha transacción. Tener en cuenta los posibles errores que puedan surgir.

2- Diagrama de clases

Se necesita modelar una capa de usuarios en donde cada usuario puede tener permisos exclusivos para utilizar determinados servicios del sistema. El usuario con perfil “superusuario” podrá cargar usuarios y perfiles y asignar responsabilidades; entre ellas puede asignar la responsabilidad de crear usuarios y de asignar responsabilidades. Se pide un diagrama de clases que solucione este requisito.