

ESTRUCTURAS DE DATOS Y ALGORITMOS II

QUINTA ENTREGA DE EJERCICIOS

Se aceptan entregas por Aulas hasta el 13/11/2016 a las 23:50

Para la realización de este trabajo la cátedra provee un proyecto de Visual Studio con las funciones a implementar, algunos datos de prueba y el framework visto en clase.

IMPORTANTE:

1. La definición de cada función incluye siempre indicar sus pre y postcondiciones.
2. Las pruebas provistas por la cátedra constituyen meros ejemplos y no son para nada exhaustivas. Se espera que los alumnos desarrollen pruebas propias para su código.
3. Debe entregarse solamente una carpeta comprimida conteniendo el proyecto "Obligatorio", sin el Framework.

?1. Transponer Matriz.

Juan se encuentra cursando Álgebra Lineal, curso en el que estudia matrices de dos dimensiones. Debido a que tiene dificultad para identificar si sus ejercicios fueron resueltos correctamente, le pidió a usted que diseñe un algoritmo que, dada una matriz de dimensiones $n \times n$, donde n es un natural de la forma 2^k , calcule la traspuesta de dicha matriz. La complejidad de dicho algoritmo debe ser $O(n^2 \log_2 n)$.

?2. Problema de la Silueta.

Usted se enfrenta a la siguiente situación: dada una vista horizontal de una ciudad que consta únicamente de edificios rectangulares, le piden diseñar un algoritmo que dada la ubicación y altura exactas de los edificios de la ciudad, halle la "silueta" o línea de corte de todos los edificios con el cielo; es decir, la silueta de todos los edificios juntos, pero eliminando las intersecciones ocultas entre los mismos. Dicho algoritmo debe ejecutar en un orden promedio de $O(n \log_2 n)$ siendo n la cantidad de edificios.

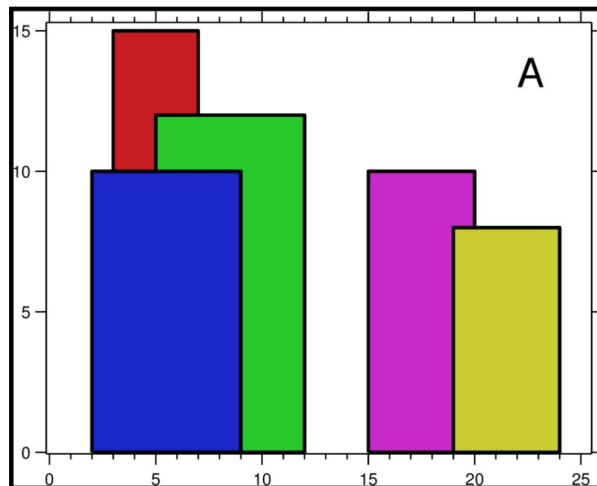
La entrada consta de la información sobre cada edificio. Dicha información está representada como una tripleta de la siguiente forma: $(x_{inicial}, x_{final}, h)$ donde:

- $x_{inicial}$ y x_{final} representan la base del edificio (desde donde arranca hasta donde termina)
- h representa la altura del mismo.

Por ejemplo una posible entrada con cinco edificios puede ser la siguiente:

[[2 9 10], [3 7 15], [5 12 12], [15 20 10], [19 24 8]]

La cual representaría la siguiente vista horizontal:

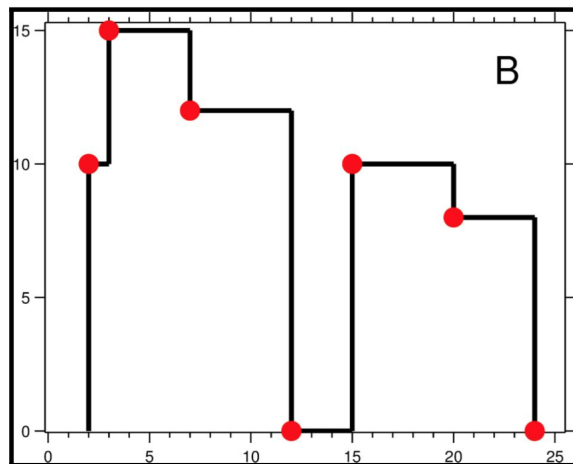


Finalmente, el algoritmo debe retornar un conjunto ordenado de “tiras”. Una tira es entendida como una pareja (x, h) donde:

- x representa la posición de la tira.
- h es la altura de la tira.

Dichas tiras, representan cada una de las líneas verticales que definen a la silueta. ¿Cómo se dibuja una tira? Una tira arranca siempre desde la posición x correspondiente y a una altura y la cual es la altura en la que terminó su tira inmediatamente anterior. Luego se dibuja la tira hasta que esta alcance su altura h indicada, midiendo desde el piso (lo cual implica que se pueda tanto crecer como decrecer en altura).

Notar que la ultima tira, aquella que define donde el edificio termina, simplemente se usa para marcar la terminación de la silueta, y obviamente tiene que tener siempre altura cero. Además, el suelo entre dos edificios adyacentes debería ser considerado como parte del contorno de la silueta. Para la entrada mostrada anteriormente, la salida debería ser la siguiente:

**Notas:**

- En la figura de la silueta, se marcan con rojo aquellos puntos donde finalizan las tiras.
- Las posiciones $x_{inicial}$, x_{final} de los edificios de entrada están ordenadas en orden ascendente, siempre de izquierda a derecha.
- La salida deberá estar ordenada ascendentemente en la coordenada x de cada tira.
- No pueden haber líneas horizontales consecutivas en la salida. Por ejemplo:

$[\dots [2 \ 3], [4 \ 5], [7 \ 5], [11 \ 5], [12 \ 7] \dots]$

La anterior no es una salida de una silueta aceptable. Las tres tiras de altura 5 deben ser unidas en una sola de la siguiente forma:

$[\dots [2 \ 3], [4 \ 5], [12 \ 7], \dots]$

Sugerencias:

¿Cómo haría para representar la silueta de un solo edificio?

¿Existe alguna diferencia entre unir una silueta con otra y unir un edificio con una silueta? ¿La respuesta a la pregunta anterior le sugiere alguna técnica de diseño de algoritmos para el problema en cuestión?

?3. Coeficientes Binomiales

Un amigo suyo, Álvaro, está estudiando combinatoria y le propone lo siguiente: diseñe un algoritmo que, dado dos naturales n y k , calcule el número de todas las agrupaciones posibles de k elementos que pueden hacerse con un conjunto con n elementos. Álvaro dice que no considera diferentes a agrupaciones de mismos elementos pero diferente orden, es decir que nunca se van a repetir elementos y que siempre se debe cumplir que $n \geq k$. Por otro lado, le pide que cumpla los siguientes ordenes:

- $O(n)$ en espacio (la memoria requerida para ejecutar el algoritmo debe ser n , es decir, el espacio memoria que ocupan todas las variables propias del algoritmo no puede superar a n).
- $O(nk)$ en tiempo de ejecución.

Para ayudarlo un poco, Álvaro le habla sobre el famoso “Triángulo de Pascal” (aunque según él fue inventado por el “Maestro Tartaglia”) y le brinda la siguiente recurrencia:

$$\binom{0}{0} = 1$$
$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}$$