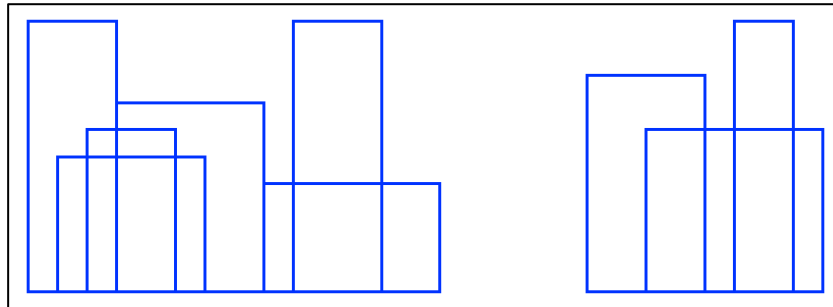


EI1022/MT1022 - Algoritmia

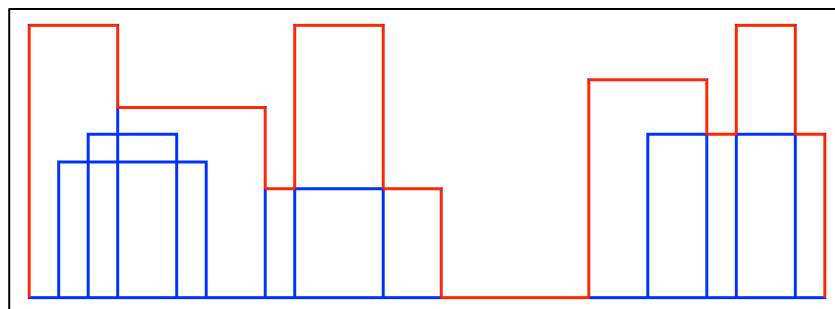
Divide y Vencerás

Entregable 4 - Fecha de entrega 11 de diciembre de 2019

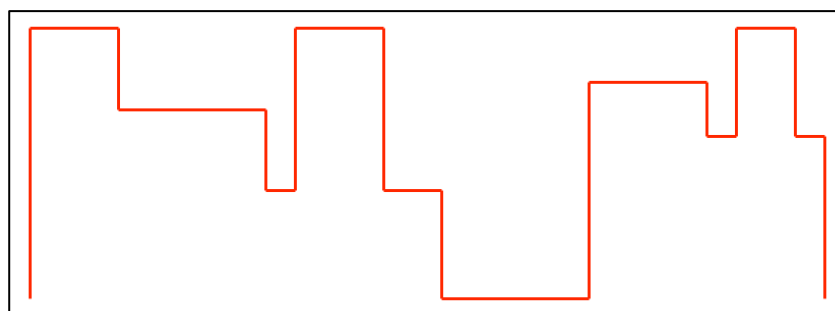
Queremos obtener el *skyline* de una ciudad representada a partir de los perfiles de sus edificios. Por ejemplo, a partir de los edificios siguientes:



Podemos marcar la altura máxima en cada abscisa:



Y obtenemos este perfil:



Representación de los datos

Podemos representar cada edificio mediante una tupla de tres enteros: abscisa izquierda, altura y anchura. La ciudad consistiría en una lista de esas tuplas. La lista de nuestro ejemplo es $[(1, 10, 3), (2, 5, 5), (3, 6, 3), (4, 7, 5), (10, 10, 3), (9, 4, 6), (20, 8, 4), (22, 6, 6), (25, 10, 2)]$.

El perfil lo representaremos mediante una lista de enteros $[x_0, h_0, x_1, h_1, \dots, x_{n-1}, h_{n-1}, x_n]$ en la que x_0 es la abscisa del inicio del perfil, h_0 es la altura desde x_0 hasta x_1 . Y así sucesivamente. Para minimizar la longitud de la salida, debe cumplirse que $h_{i-1} \neq h_i$ para $0 < i < n$.

En nuestro ejemplo, el perfil es $[1, 10, 4, 7, 9, 4, 10, 10, 13, 4, 15, 0, 20, 8, 24, 6, 25, 10, 27, 6, 28]$.

Se pide

- a) Un **programa** de línea de órdenes (entregable4.py) que reciba como parámetro el nombre de un fichero de texto en el que cada línea se corresponde con un edificio representado mediante tres enteros separados por blancos. El fichero de entrada del ejemplo¹ sería:

```
1 10 3
2 5 5
3 6 3
4 7 5
...
```

Como resultado de su ejecución el programa mostrará el perfil como una línea de texto formada por enteros separados por blancos. En nuestro ejemplo:

```
1 10 4 7 9 4 10 10 13 4 15 0 20 8 24 6 25 10 27 6 28
```

El coste de ejecución del programa debe ser $O(n \log n)$, donde n es el número de edificios.

- b) **Memoria** del entregable. Una memoria en la que se detallen los pasos seguidos y dificultades experimentadas, a la vez que se presente una solución para el problema en cuestión. Indica los costes espacial y temporal de tu algoritmo. Las faltas de ortografía penalizan; una redacción descuidada penaliza.
- c) **Actas** de las reuniones mantenidas hasta la entrega. Recordad que uno de vosotros será el secretario y se encargará de tomar nota en la/s reunión/es de trabajo. El cargo de secretario es rotativo: será una persona diferente para cada entregable.
- d) **Valoración personal**. Cada miembro del grupo deberá escribir una breve valoración del trabajo realizado y de los resultados obtenidos.

Como respuesta a la tarea correspondiente en el aula virtual se enviará un fichero comprimido con todo (a, b, c y d). Además, los apartados b, c y d se entregarán también en papel, en clase, en la fecha indicada.

Detalles de implementación

Para conseguir el coste pedido, puedes utilizar el esquema de 'Divide y vencerás'. Divide la lista de edificios en dos partes, calcula recursivamente el *skyline* de cada una y combínalos.

Prueba del programa

En el aula virtual puedes descargar el paquete 'test_entregable4.zip'. Este paquete contiene un directorio pruebas con 32 problemas (ficheros ciudad_{num1}_{num2}.i) y sus soluciones (ficheros ciudad_{num1}_{num2}.o). En el nombre de cada archivo, {num1} indica el número de edificios del

¹ Este fichero, ejemplo.i, y su salida, ejemplo.o, se encuentran en auxiliares-entregable4.zip.

problema. Además, en este directorio también encontrarás el problema `ejemplo.i` y su solución `ejemplo.o`.

No se podrá obtener una nota superior a 4 en el entregable si no se superan todas las pruebas del paquete en menos de un minuto. Tras la entrega, se realizarán pruebas adicionales cuyos resultados se tendrán en cuenta para la calificación del entregable.

Extensión opcional

Modificar el programa `'entregable4.py'` para que si recibe un segundo parámetro con el valor `'-g'` muestre gráficamente los edificios y el *skyline* como en la segunda imagen de nuestro ejemplo.

Para facilitar la realización de esta extensión podéis utilizar la clase `SkylineViewer` que se encuentra disponible en el fichero `'test_entregable4.zip'`. El fichero `demo_skylineviewer.py` contiene un ejemplo de utilización de la clase `SkylineViewer`.

Ejemplo de utilización:

```
from skylineviewer import SkylineViewer

buildings = [(1, 10, 3), (2, 5, 5), (3, 6, 3), (4, 7, 5), (10, 10, 3),
             (9, 4, 6), (20, 8, 4), (22, 6, 6), (25, 10, 2)]

skyline = [1, 10, 4, 7, 9, 4, 10, 10, 13, 4, 15, 0, 20, 8, 24, 6, 25,
           10, 27, 6, 28]

viewer = SkylineViewer(skyline)

for b in buildings:
    viewer.add_building(b)

viewer.run()
```

Plazos de entrega

- Fecha de entrega en el aula virtual: 11 de diciembre de 2019.
- Fecha de entrega de la memoria: 16 de diciembre de 2019, en clase.

No os quedéis colgados. Recordad que hay tutorías.