

# ALGORITMOS: AYUDANTÍA 6

Ayudante: Yerko Ortiz

*"Good abstractions turn a nearly impossible task into two manageable ones. The first one is defining and implementing the abstraction. The second is using these abstractions to solve the problem at hand."*

Andrew Tanenbaum, Modern  
Operating Systems.

**Objetivo de la ayudantía:** Resolver problemas de divide and conquer y un ejemplo de ordenamiento de strings

## Vista horizontal

Dadas las dimensiones y posiciones de  $N$  rectángulos en un plano de dos dimensiones, compute la vista horizontal del plano, donde la vista horizontal del plano se define como la figura o conjunto de figuras geométricas con área máxima que no contienen líneas en su interior.

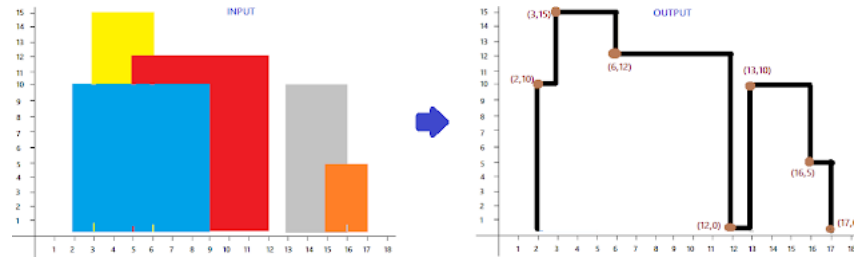


Figura 1: Ejemplo gráfico de la vista horizontal

## Formato de la entrada

- La primera línea contiene un entero  $N$  que representa la cantidad de rectángulos.
- Las  $N$  líneas subsiguientes contienen tres enteros  $l, r, h$  que representan la posición a más a la izquierda en el plano, la posición más a la derecha y la altura del rectángulo (todos los rectángulos empiezan desde la altura 0).

## Dominio de la entrada

- $1 \leq N \leq 10^6$
- $1 \leq l, h, r \leq 10^3$

## Formato de la salida

- La lista de vertices de la figura o conjunto de figuras que representan la vista horizontal.

## Casos de prueba

### ■ Entrada:

```
3
1 3 2
2 4 3
4 6 5
```

### ■ Salida:

```
1 2
2 3
4 0
4 5
6 0
```

---

## Máximo prefijo en común

Dado un conjunto de  $N$  palabras, compute el máximo prefijo que tienen en común todas las palabras del conjunto. El prefijo de un string se define como un substring de caracteres adyacentes, donde el primer caracter del substring corresponde al primer caracter del string. Por ejemplo sea la palabra banana, se dice que banan es un prefijo de banana, así mismo ban es prefijo de banana, pero por otro lado anana no es prefijo de banana.

## Formato de la entrada

- La primera linea contiene un entero  $N$  que representa la cantidad de palabras.
- Las siguientes  $N$  lineas contienen  $N$  strings que representan la palabra  $S_i$ .

## Dominio de la entrada

- $2 \leq N \leq 10^3$
- $1 \leq |S_i| \leq 100$

## Formato de la salida

- Una linea que imprima el substring que corresponde al prefijo en común de largo máximo que las palabras tienen en común.

## Caso de prueba

### ■ Entrada:

```
5
banana
barco
bardo
banal
baneado
```

### ■ Salida:

```
ba
```

---

# Diccionario de palabras

Franco tiene un sueño - el quiere crear su propio diccionario; esta no es una tarea fácil ya que el número de palabras que conoce, no es suficiente. Para compensar las pocas palabras que conoce piensa en una brillante idea: crear un diccionario usando las palabras de algún libro en la biblioteca de su hermano. Su diccionario contendrá las palabras del libro ordenadas alfabeticamente y sin palabras repetidas.

Su tarea como ninja de la programación es diseñar e implementar un programa en java, que simule el proceso que Franco realiza para crear su diccionario.

**Nota:** su programa debe ser case insensitive(tomar todas las letras mayúsculas o minúsculas como si fuesen las mismas letras), es decir que si por ejemplo su programa lee la palabra "naranja" la considera igual a la palabra "NaraNja".

## Entrada

- La entrada consiste en un texto T con no más de 5000 lineas, cada linea tiene a lo más 200 caracteres. El input termina con EOF.

## Dominio

- $0 \leq |T| \leq 10^6$

## Salida

- Las palabras del diccionario ordenadas de forma alfabética, escritas en minúsculas y sin palabras repetidas.

## Caso de prueba

- **Entrada:**  
Adventures in park. Two guys were going to the park when they came to a fork in the road. The sign read:  
"park Closed." So they went home.
- **Salida:**  

a	
adventures	
came	road
closed	sign
fork	so
going	the
guys	they
home	to
in	two
park	went
read	were
	when

---

Gracias por su atención!