

G16

- Eric RuizDiaz

- Luis Pereira

## COSTO ASINTOTICO

### Explicación de algoritmo:

Este algoritmo que encuentra el árbol de expansión máxima usa el método de Kruskal para árbol de expansión mínimo, pero modificado para obtener las aristas de mayor peso en orden decreciente.

### Los pasos son:

$O(V)$	{ - Recorre todos los vértices para hacer las conexiones de aristas
$O(E \log(E))$	{ - Ordena las aristas en orden decreciente
$O(V \log(V))$	{ - Por cada arista en orden decreciente revisa si los dos vértices están ya están pareados, si no es así, entonces se cargan al árbol

### Calculo de tiempo:

$$T(n) = O(V) + O(E \log(E)) + O(V \log(V))$$

$$T(n) = O(E \log(E)) + O(V \log(V))$$

$$|E| \leq |V|^2$$

$$\log(E) \leq \log(V^2)$$

$$\log(E) \leq 2 \log(V)$$

**El costo asintótico es de:  $O(E \log(V))$**

## TAD UTILIZADOS

- Para almacenar las aristas se utiliza una clase llamada Arista que almacena el vértice inicial, el vértice final, y el tamaño de la arista
- Para almacenar el grafo se utiliza una matriz de adyacencia
- Para almacenar el árbol de expansión máxima se utiliza una matriz
- Un stack para la ver si existen ciclos en el grafo
- Un vector donde se almacenan los vértices visitados