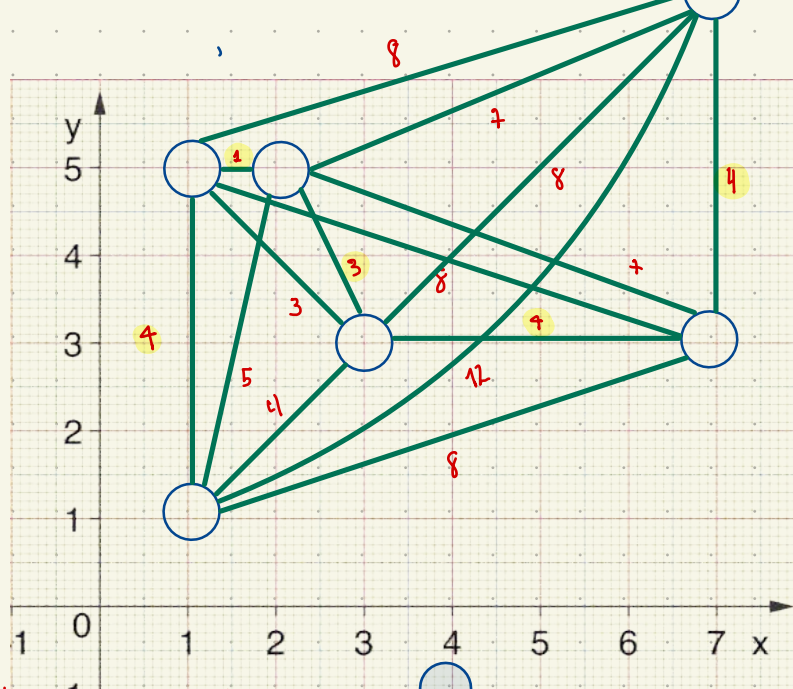


# Tarea 3

Sebastián Achondo Silva

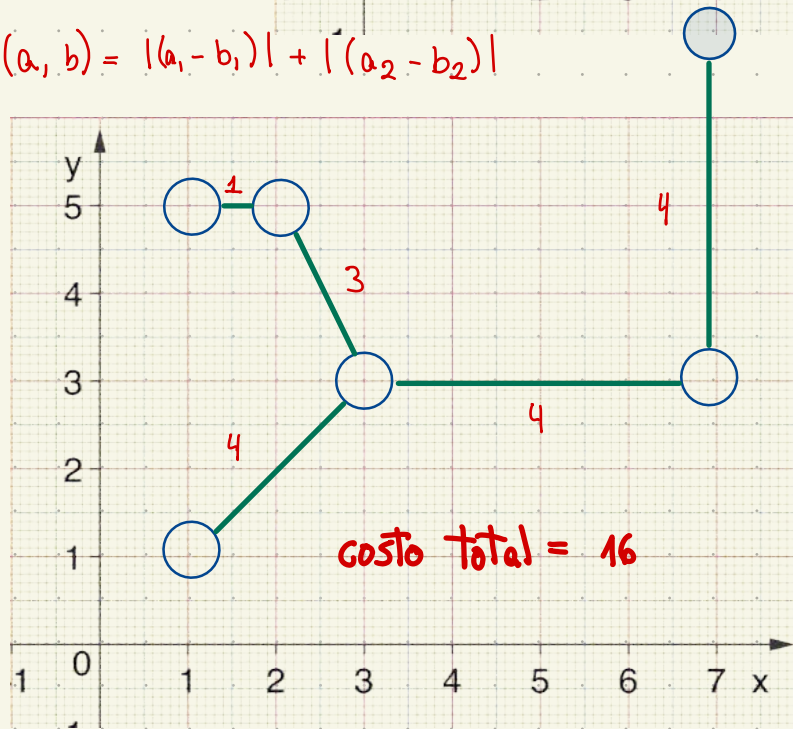
## Parte 2: MST

Ej input:  
Ojo, es grafo  
completo

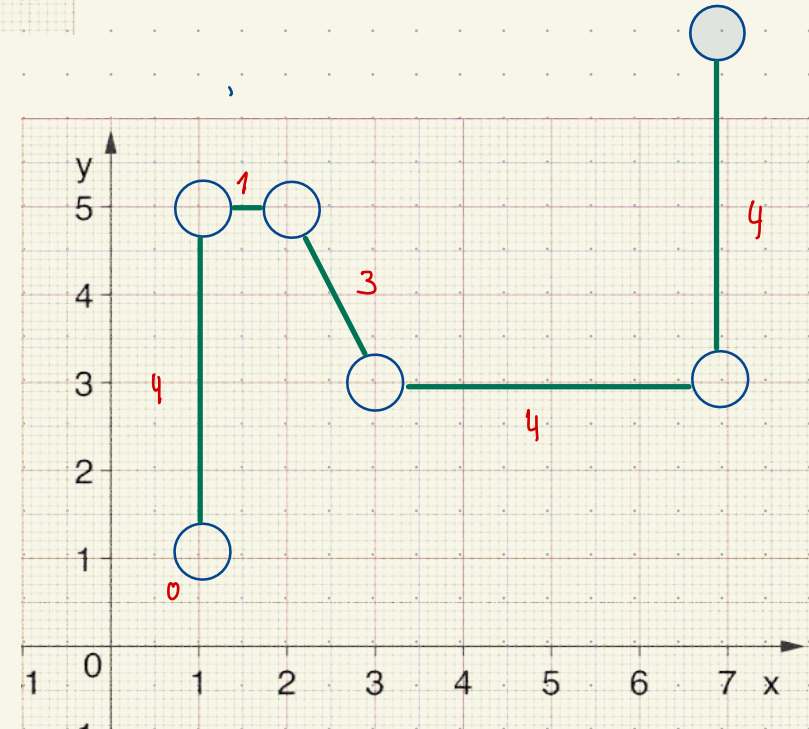


$$d(a, b) = |a_1 - b_1| + |a_2 - b_2|$$

output 1:



Output 2:  
mejorado



Para mejorarlo entonces se busca disminuir el num  
de aristas por nodo!

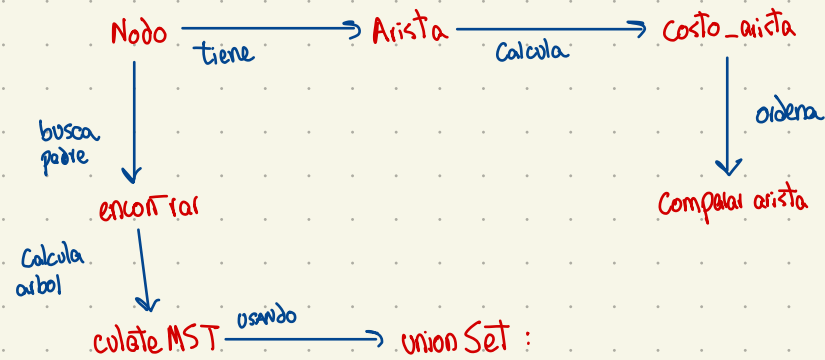
1) Analisis enunciado: El programa pide, a partir de un arbol completo determinen un MST tal que el costo sea minimo y el grado de los nodos tmb.

2) Planificación:

```
struct Nodo { x, y }
struct Arista { src, dest, costo }
```

costo\_arista: Calcula el costo segun manhattan  
comparar\_arista: Para ordenar aristas con qsort  
encontrar: Dado un nodo, encuentra su padre  
unionSet: Une dos conjuntos disjuntos.  
calculateMST: Calcula un Arbol MST con el algoritmo Kruskal

3) Diagrama



## Parte 1 : backpack

1) Analisis enunciado: El programa pide generar un algoritmo que distribuya todos los items dados en ciertos agentes que tienen una capacidad fija. Al final hay que minimizar el numero de agentes pero maximizando la cantidad de elementos que lleva cada caba uno sin violar las restricciones de peso.

2) Planificación: `struct Item (id, peso)`

`crear Items`: Crea un arreglo del tamaño de items y devuelve puntero a arreglos.

`comparar Pesos`: A traves de `qsort` compara los pesos y los ordena en descendencia.

`destruir Items`: Libera la memoria del arreglo de punteros.

`asignar Mochilas`: La encargada de asignar los elementos a las mochilas de manera optima.

3) Diagrama

