

Árbol recubridor mínimo

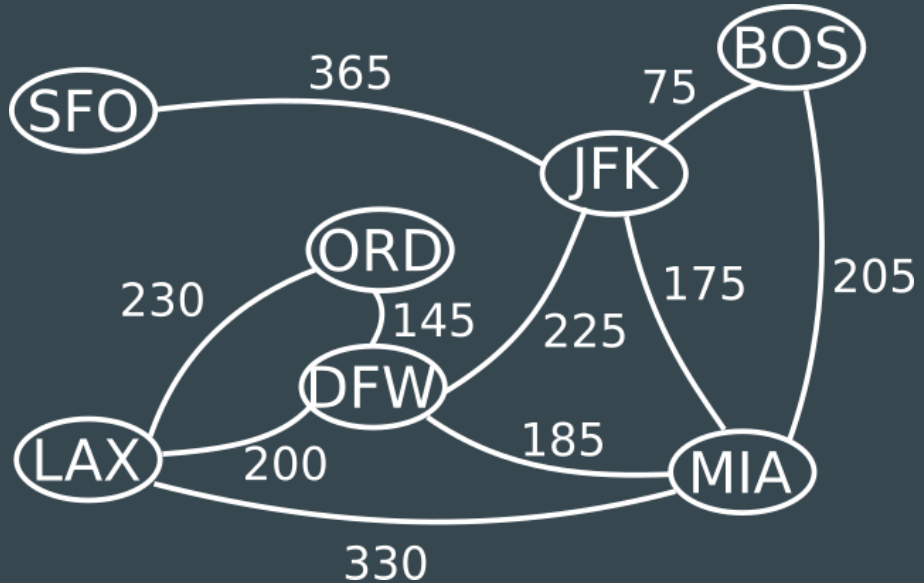
...

Algoritmos y Estructuras de Datos
2024

Se desean conectar en red todas las computadoras de un aula ¿Cómo podría calcular el orden en que deberían conectarse para utilizar la menor cantidad de cable posible?

Árbol recubridor mínimo

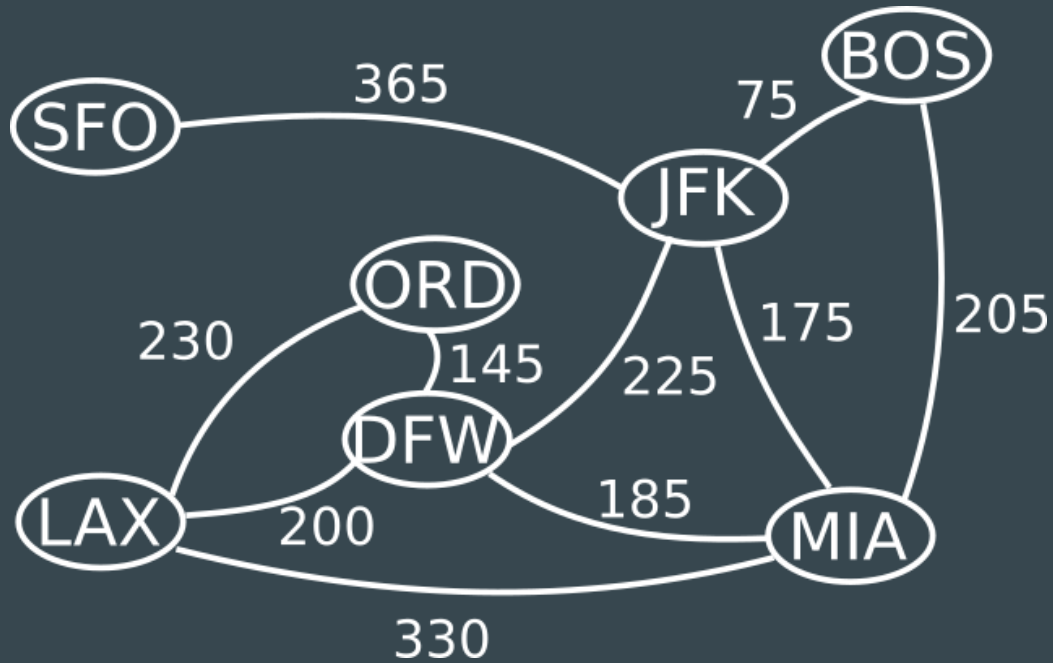
- Un árbol recubridor contiene **todos los nodos** de un grafo **no dirigido** y **conectado**.
- Si el grafo posee n nodos, el árbol recubridor posee $n-1$ aristas.
- El **árbol recubridor mínimo** es el árbol con **menor peso total**.



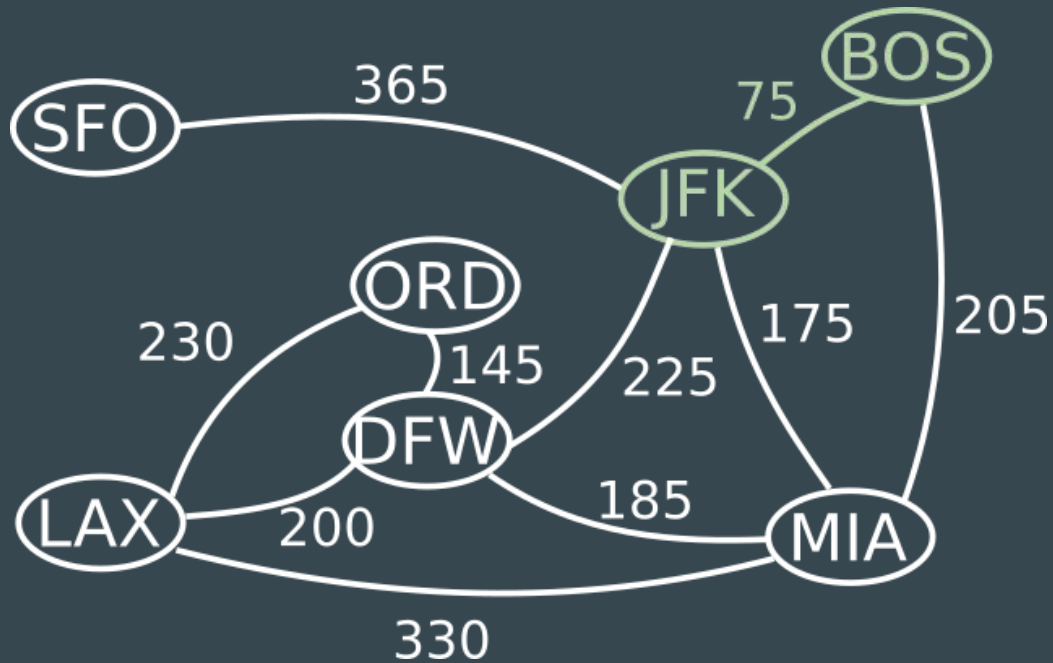
Algoritmo de Kruskal

- Es un algoritmo **ávido**.
- La **heurística** en la que se basa es seleccionar el arco de menor peso,
- A medida que se seleccionan nodos, se van formando distintos subárboles, que serán combinados en un **árbol final**.
- Si la arista de menor peso conecta nodos ya presentes en un mismo árbol, se descarta
- El algoritmo termina cuando se forma un solo árbol de $n-1$ aristas.

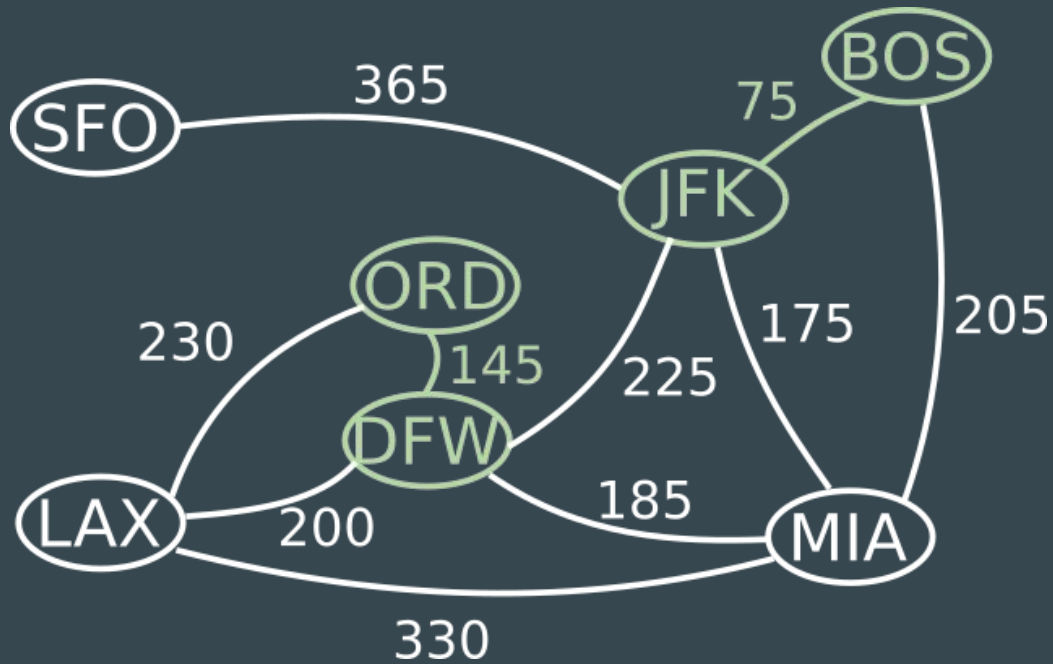
Algoritmo de Kruskal



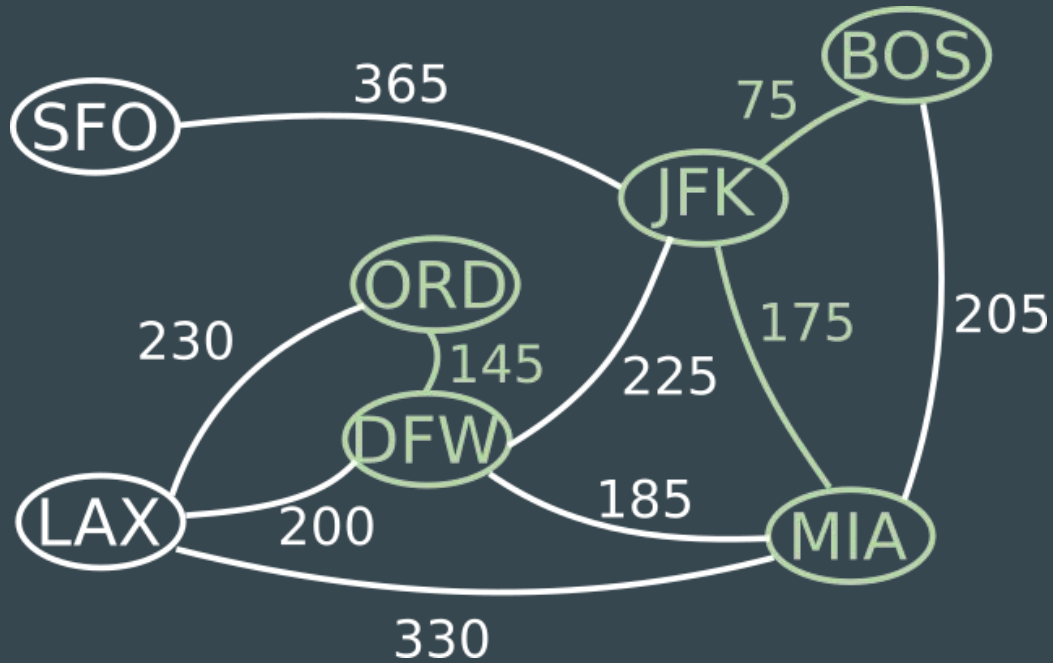
Algoritmo de Kruskal



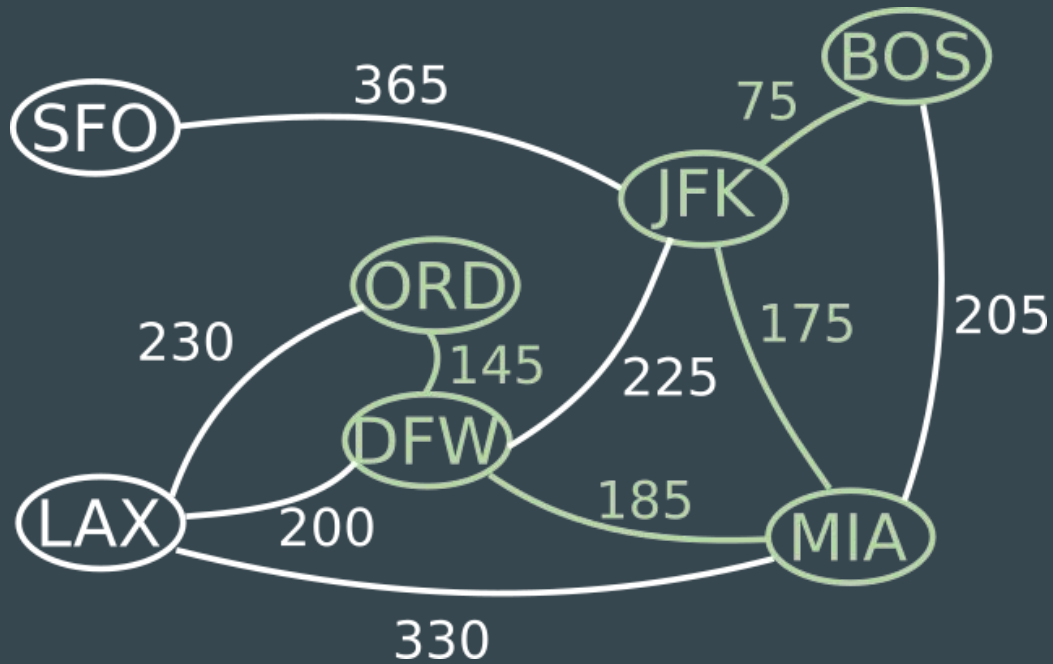
Algoritmo de Kruskal



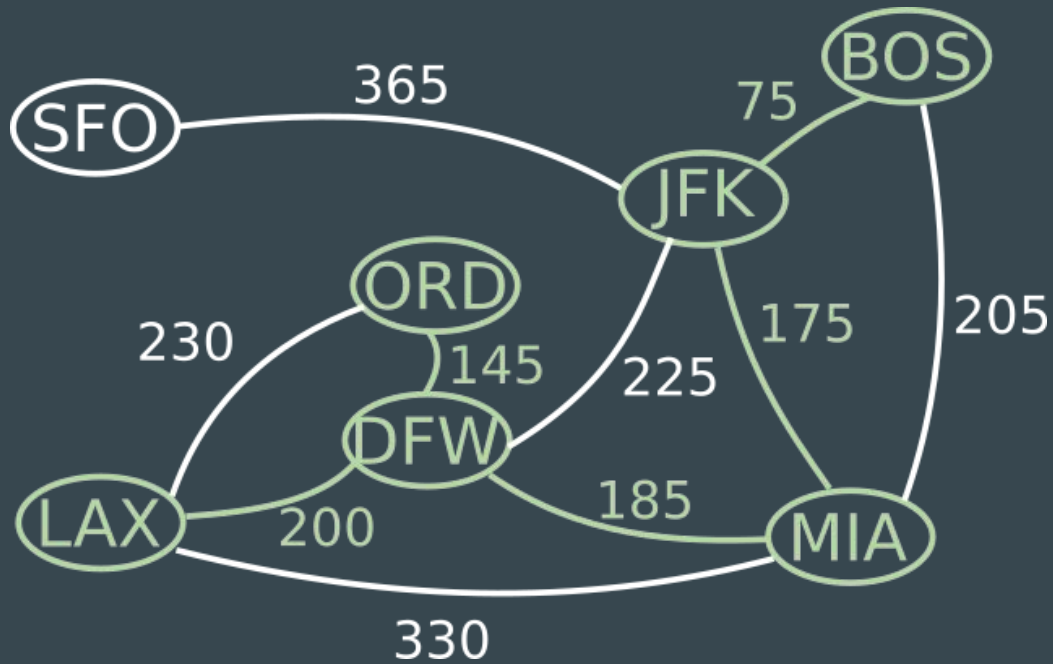
Algoritmo de Kruskal



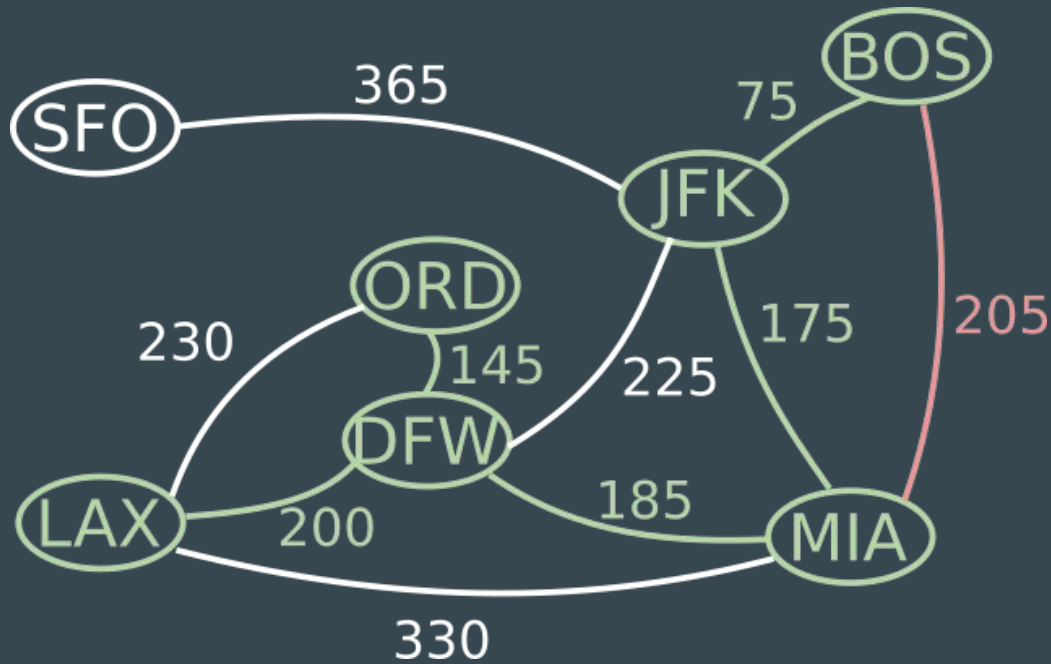
Algoritmo de Kruskal



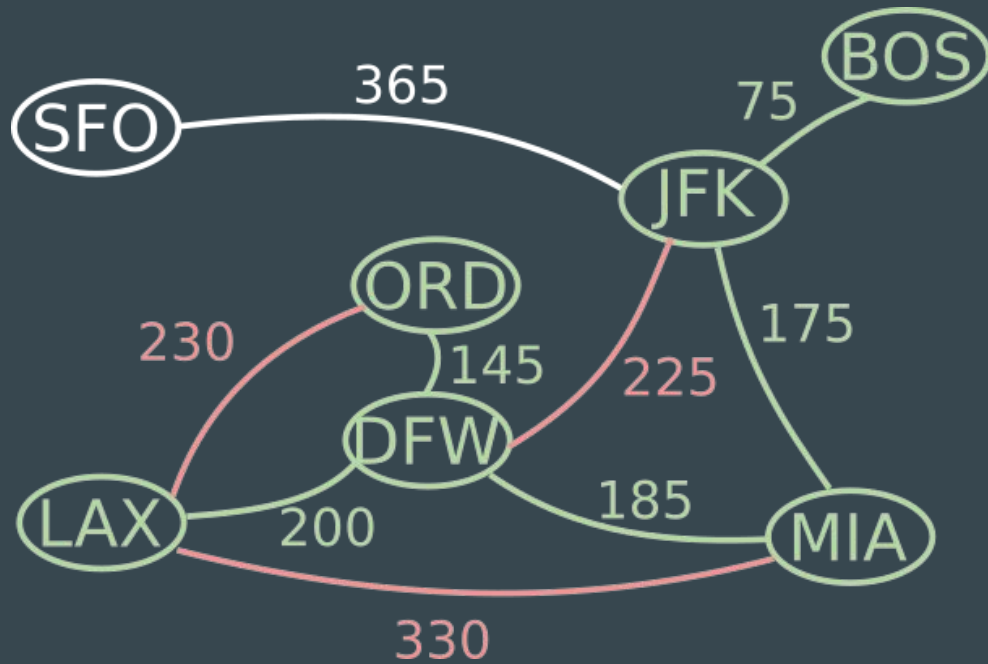
Algoritmo de Kruskal



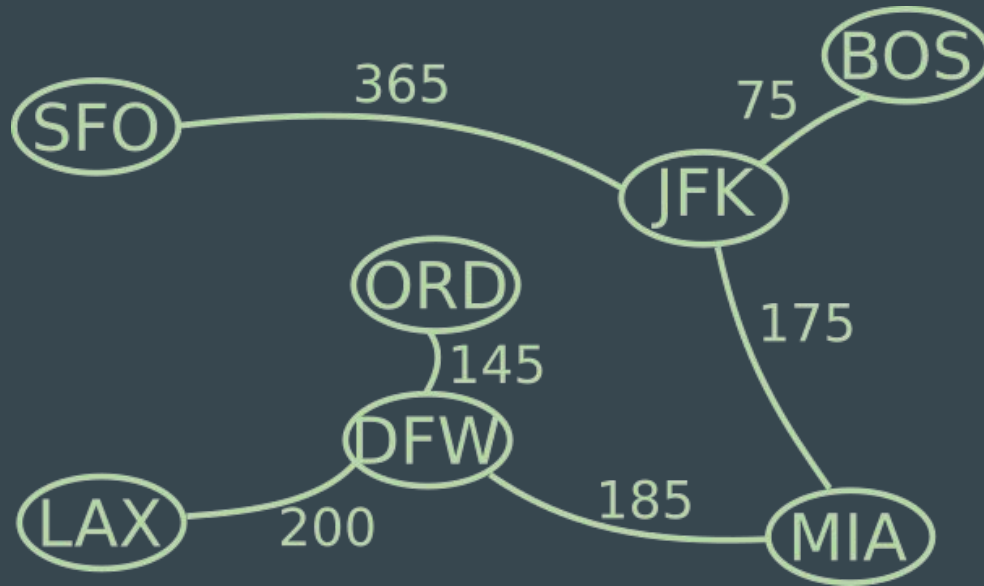
Algoritmo de Kruskal



Algoritmo de Kruskal



Algoritmo de Kruskal



Algoritmo de Kruskal: pseudocódigo

Entrada: grafo, no dirigido, ponderado

armar **tabla de pesos ordenada**

mientras que el número de aristas en el árbol mínimo $\neq n-1$:

 elegir **arista de menor peso**, quitar de la lista

 si los nodos no están en ningún **subárbol**:

 crear **subárbol**

 si están en **distintos subárboles**:

 combinar **subárboles**

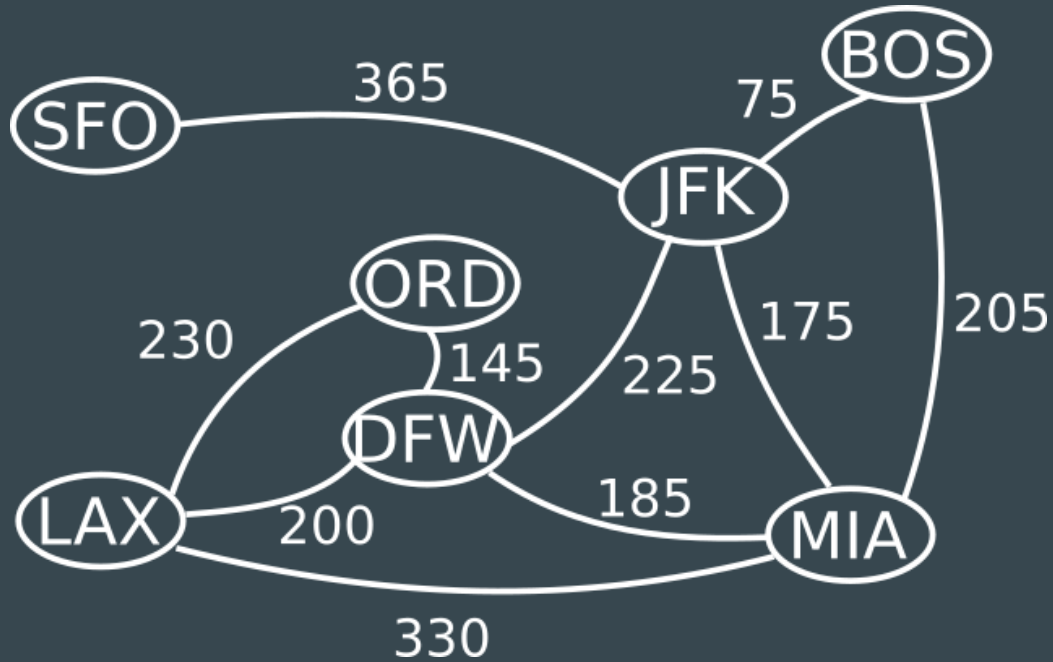
 si están en el mismo árbol:

 descartar arista

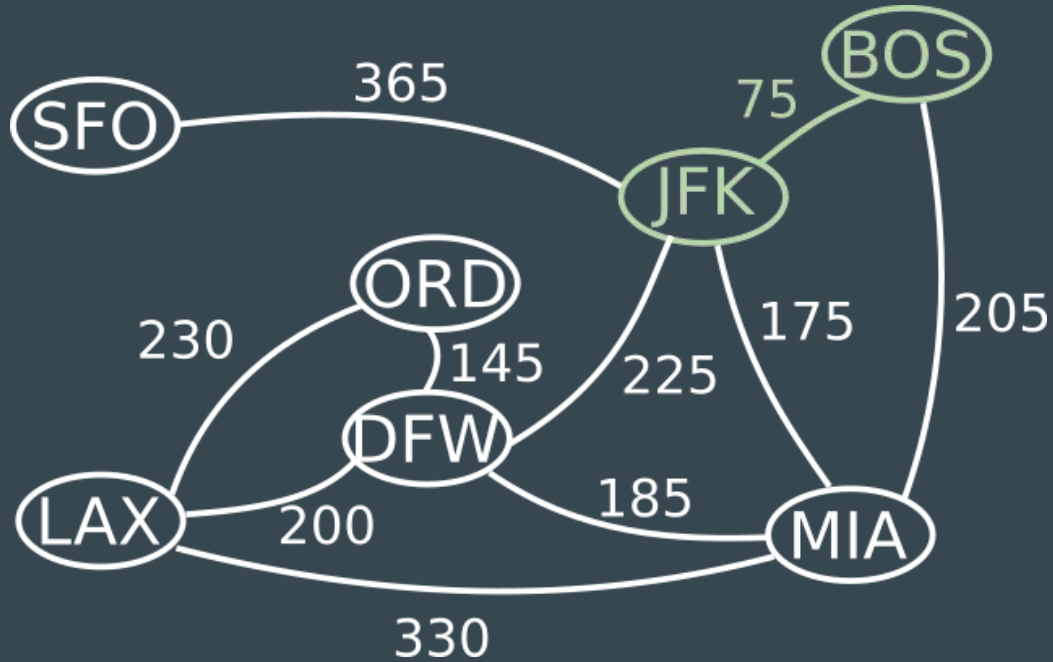
Algoritmo de Prim

- También es un algoritmo ávido.
- Se comienza eligiendo la **arista de menor peso**, y se continúa con la arista de menor peso conectada a los nodos **ya visitados**.
- Si la arista de menor peso lleva a un nodo ya visitado, se descarta.
- El algoritmo termina cuando todos los nodos fueron visitados

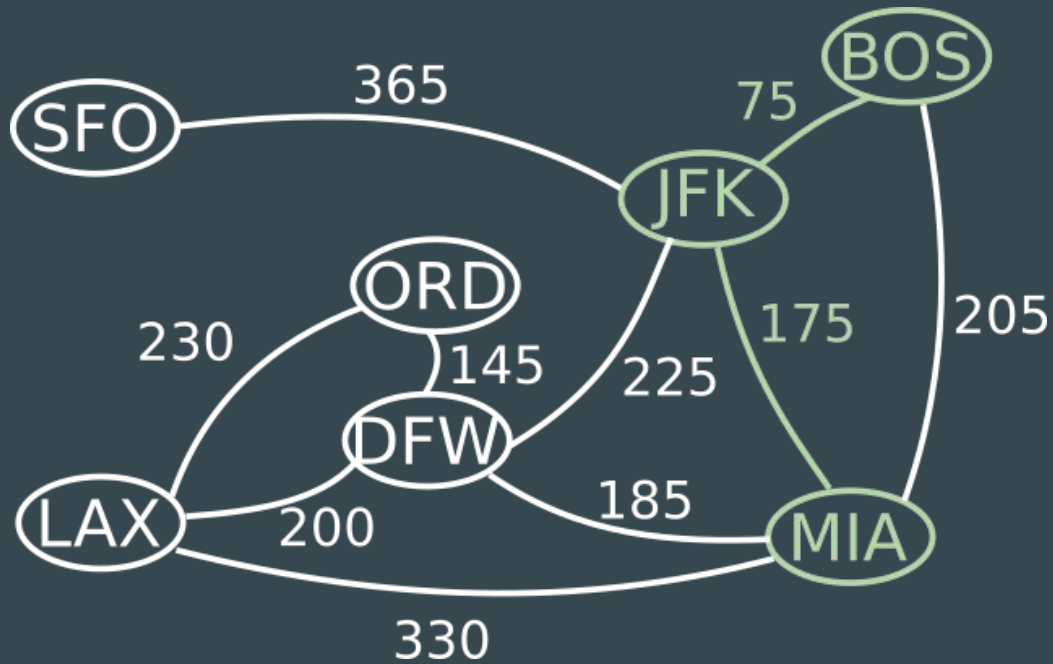
Algoritmo de Prim



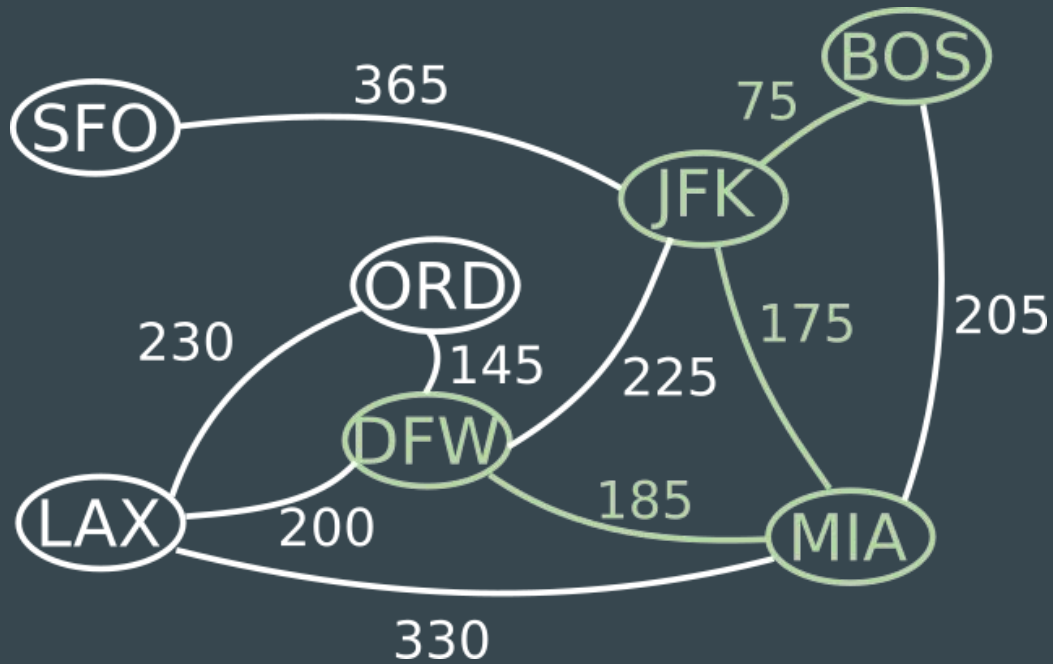
Algoritmo de Prim



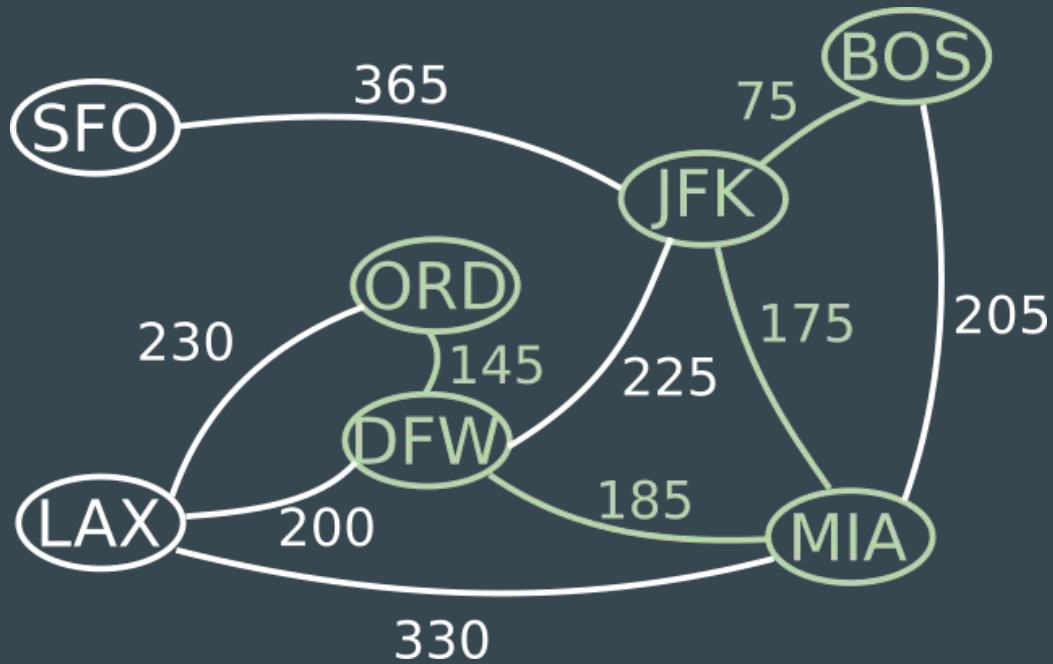
Algoritmo de Prim



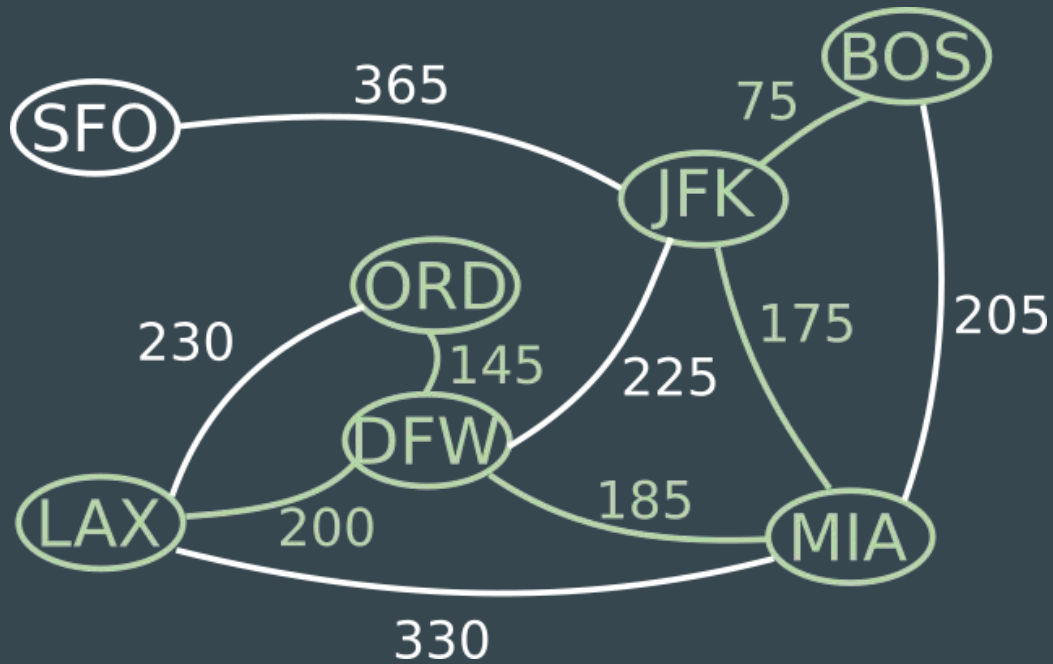
Algoritmo de Prim



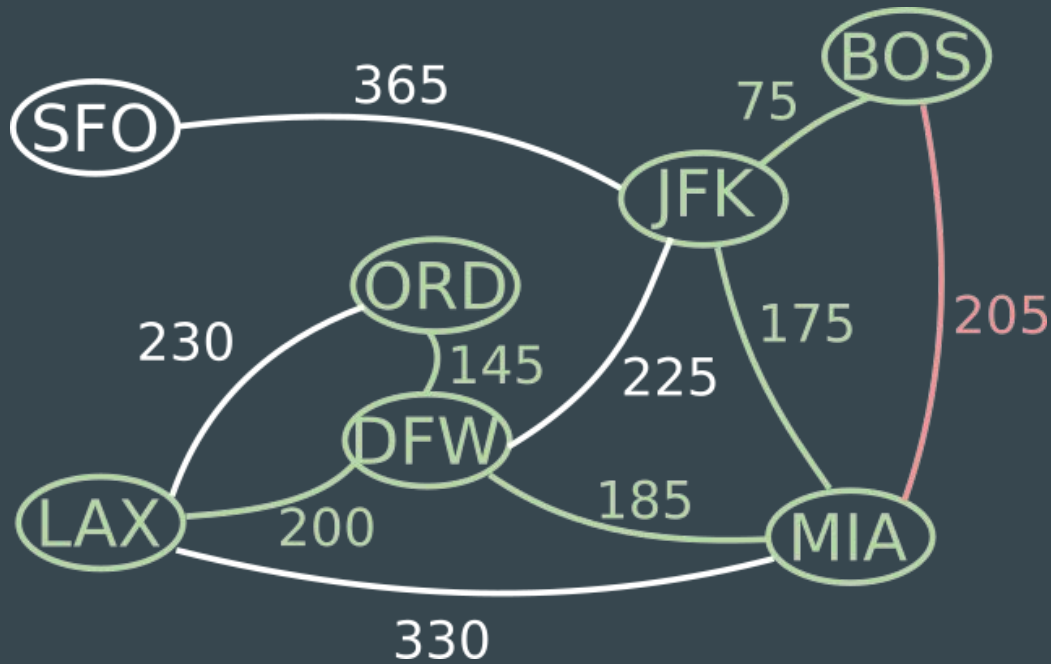
Algoritmo de Prim



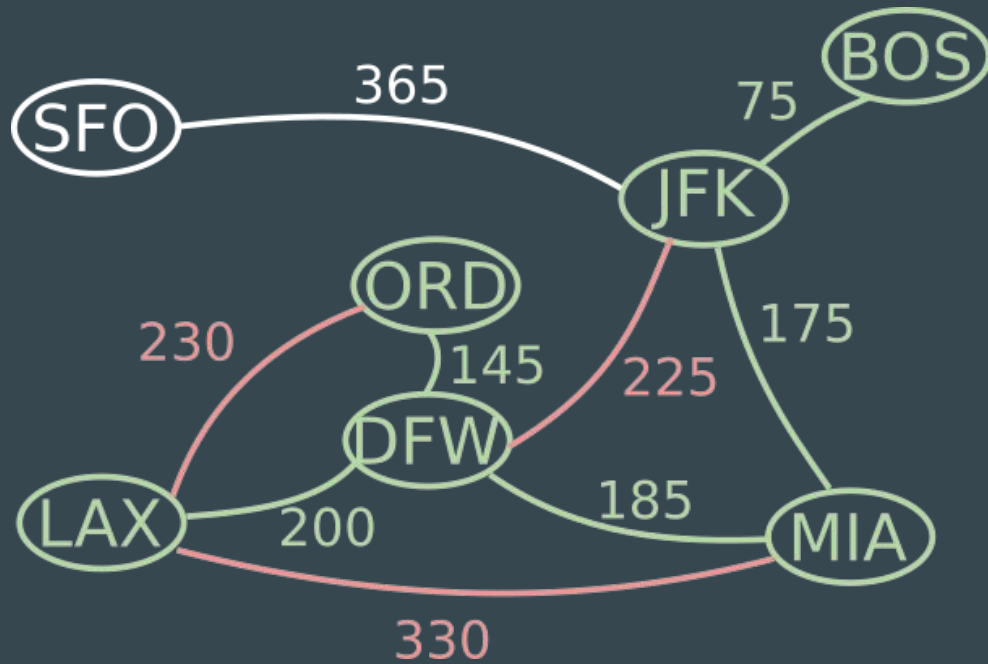
Algoritmo de Prim



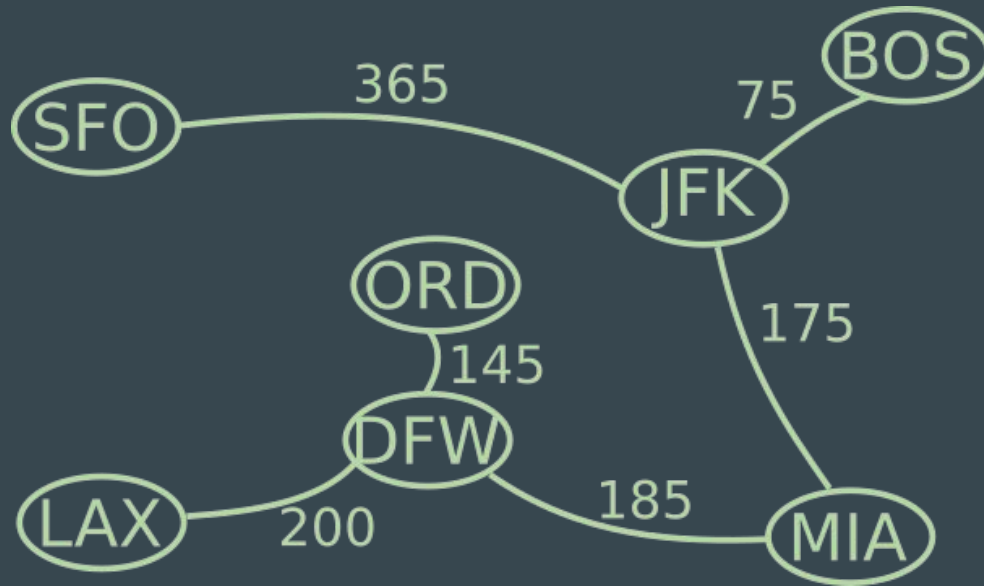
Algoritmo de Prim



Algoritmo de Prim



Algoritmo de Prim



Algoritmo de Prim: pseudocódigo

Entrada: **grafo**, no dirigido, ponderado

seleccionar **arista** de menor peso

añadir **arista** al árbol mínimo

añadir **nod**os a **visitados**

mientras $\text{len}(\text{visitados}) < \text{len}(\text{grafo})$:

 seleccionar **arista** de menor peso conectada a **nod**os **visitados**

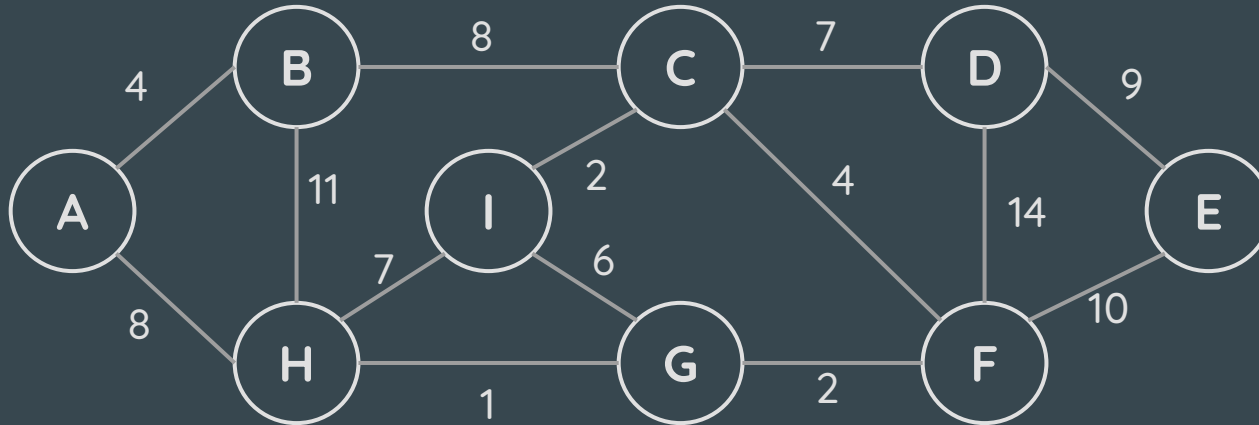
 si **nodo_destino** no está en **visitados**:

 añadir **arista** al árbol mínimo

 añadir **nodo** a **visitados**

Ejemplo

A partir del siguiente grafo, calcule los árboles recubridores mínimos usando los algoritmos de Kruskal y de Prim



Bibliografía

- Goodrich, M. T., Tamassia, R., & Goldwasser, M. H. (2013). Data structures and algorithms in Python. Capítulo 14.7
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to algorithms. MIT press. Capítulo 23