



PROGRAMACION III

TRABAJO PRACTICO

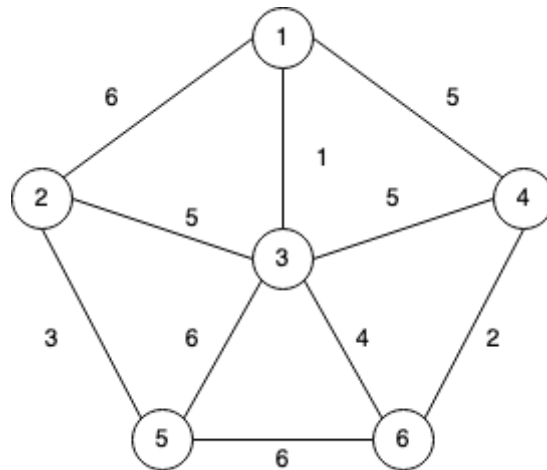
Integrantes:

- Bedecarats, Esteban LU 1107524
- Pissani, Alejandra LU 1088304

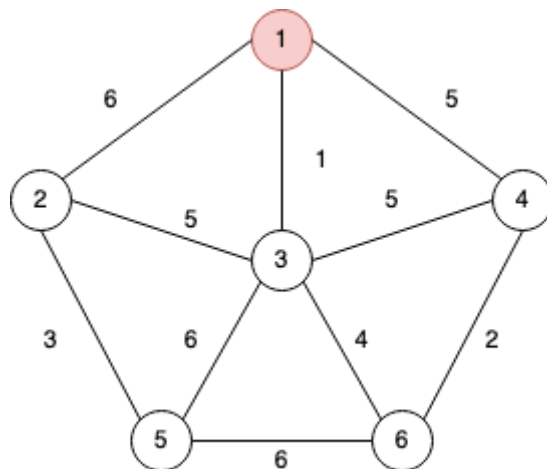
2022/MRI Pinamar

PRIM

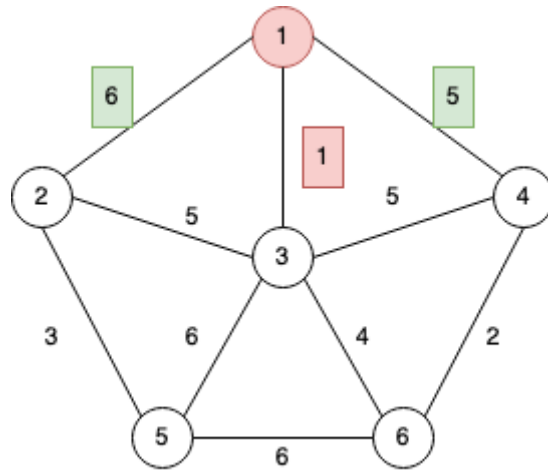
Para Prim, utilizamos como entrada este grafo:



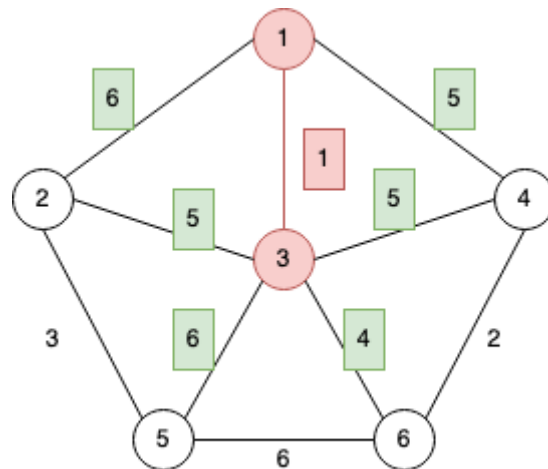
Empezamos eligiendo un nodo al azar, en este caso será el de arriba de todo **(1)** que se encuentra en la línea 103 del archivo PRIM.java.



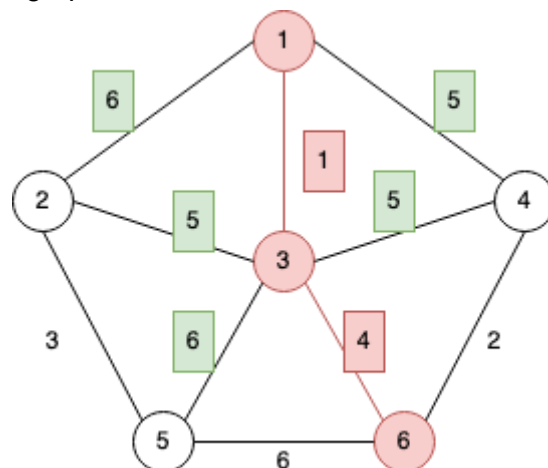
A partir de esta selección se busca el nodo con menos distancia entre sus adyacentes



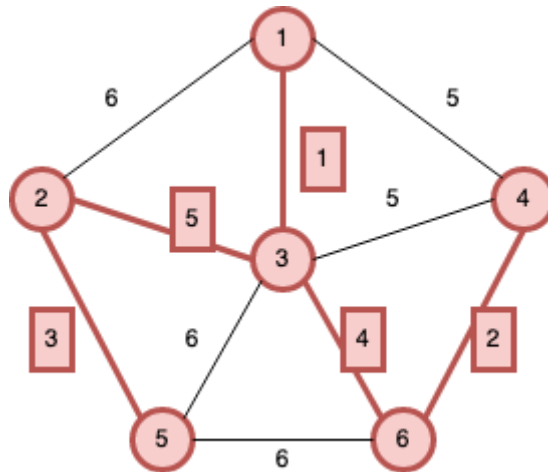
Una vez seleccionada se crea un nuevo subgrupo



A partir de cada adición, se agrega al subgrupo solución, y cada vez que se busca las distancias es desde el subgrupo entero



En este caso, entre 1 y 3 hay varios caminos entre estos y el resto pero me quedo con el camino de 3 a 6 ya que es el más corto de todos los disponibles.
Siguiendo esta lógica, el grafo termina siendo:

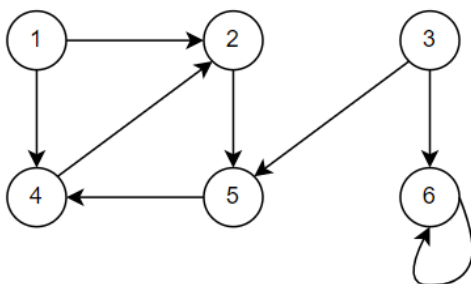


La implementación de este algoritmo está en la clase PRIM.java y la salida que se muestra es:

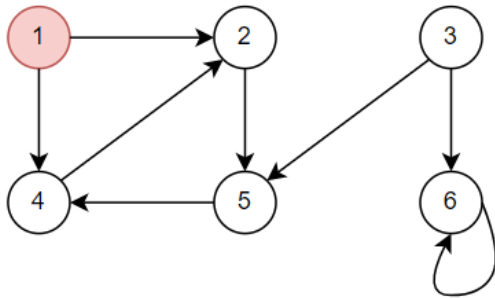
```
Prim
Origen: 1
Nodo: 5 -
Nodo: 2 - Peso: 3 -> Nodo: 5
Nodo: 4 -
Nodo: 6 - Peso: 2 -> Nodo: 4
Nodo: 3 - Peso: 5 -> Nodo: 2    Peso: 4 -> Nodo: 6
Nodo: 1 - Peso: 1 -> Nodo: 3
```

DFS

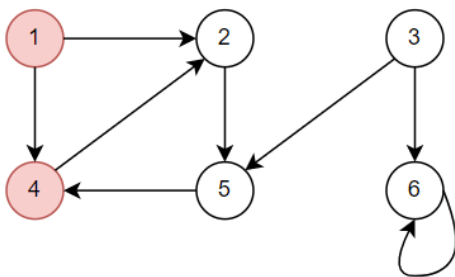
Para DFS utilizamos el siguiente grafo dirigido de entrada



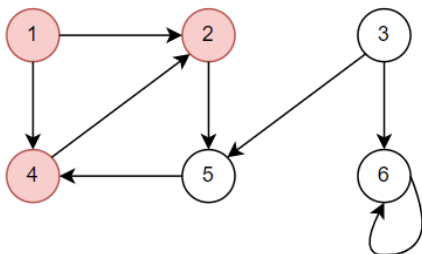
Se toma como nodo origen el **1**, que se encuentra harcodeado en la línea 53 del archivo **DFS.java** para que sea más sencillo poder variar el nodo origen y poder ver otra solución.



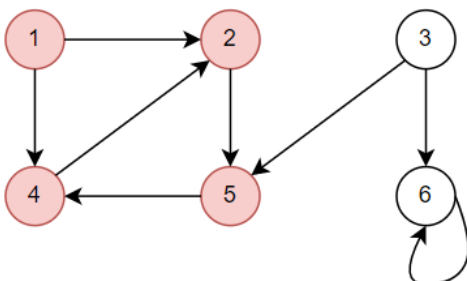
El algoritmo DFS puede elegir entre los nodos adyacentes **2** o **4** del grafo, en este caso **selecciona el nodo 4**



Luego se dirige al nodo 2



Y, finalmente, se dirige al nodo 5



Los nodos 3 y 6 quedan sin visitar ya que no son alcanzables desde el nodo 1.

La implementación del algoritmo se encuentra en DFS.java y la salida es

DFS

Origen: 1

1 -> 4 -> 2 -> 5