

Name: Alvaro Serrano
Class: CS3310
Assignment: PA6

PA6 Debugging

1. Set breakpoints in your calculateMST() method so that the program stops when the first cycle is formed. In order to show why the cycle is formed, you need to show the current edge and all the edges in the current minimum spanning tree. You can use screenshots to show the breakpoint, the condition to find the first cycle, the current edge and all the edges in the current MST.

Name	Value
find() returned	4
this	Graph (id=20)
weight	0
pq	PriorityQueue<E> (id=74)
[0]	Edge (id=42)
[1]	Edge (id=35)
parent	(id=122)
[0]	5
[1]	6
[2]	3
[3]	4
[4]	4
[5]	5
[6]	4
index	5
edge	Edge (id=43)
destination	6
source	4
weight	24
xSet	4
ySet	4

Current MST

N: 7

Parent[] = (See screenshots below)

Current edge

Source: 4

Destination: 6

Weight: 24

A cycle is formed

- Ignore the edge
- Do not add edge to spanning tree

Both vertices lie on the same set (they have the same representative, which is 4 in this case), which means that they are either directly or indirectly connected to each other with the minimum number of edges possible. Adding one more edge will again connect them creating a cycle in the minimum spanning tree that we are trying to construct.

Variables Breakpoints Expressions

NullPointerException: caught and uncaught

Graph [entry] - calculateMST()

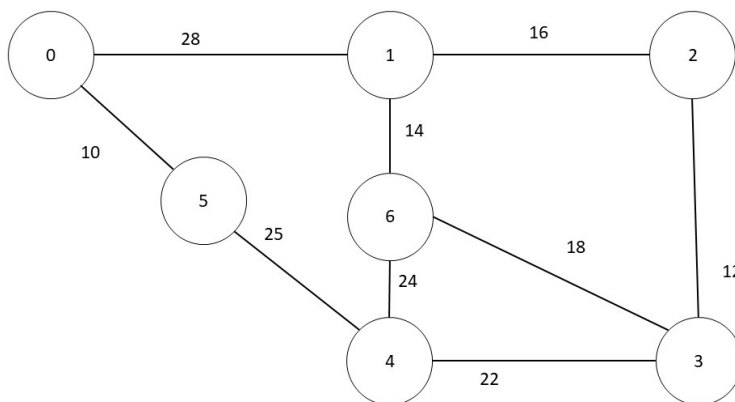
Set breakpoint on calculateMST()Before rejecting the cycle

▼ MST	ArrayList<E> (id=32)
▼ [0]	Edge (id=68)
▪ destination	5
▪ source	0
▪ weight	10
▼ [1]	Edge (id=77)
▪ destination	3
▪ source	2
▪ weight	12
▼ [2]	Edge (id=74)
▪ destination	6
▪ source	1
▪ weight	14
▼ [3]	Edge (id=71)
▪ destination	2
▪ source	1
▪ weight	16
▼ [4]	Edge (id=82)
▪ destination	4
▪ source	3
▪ weight	22

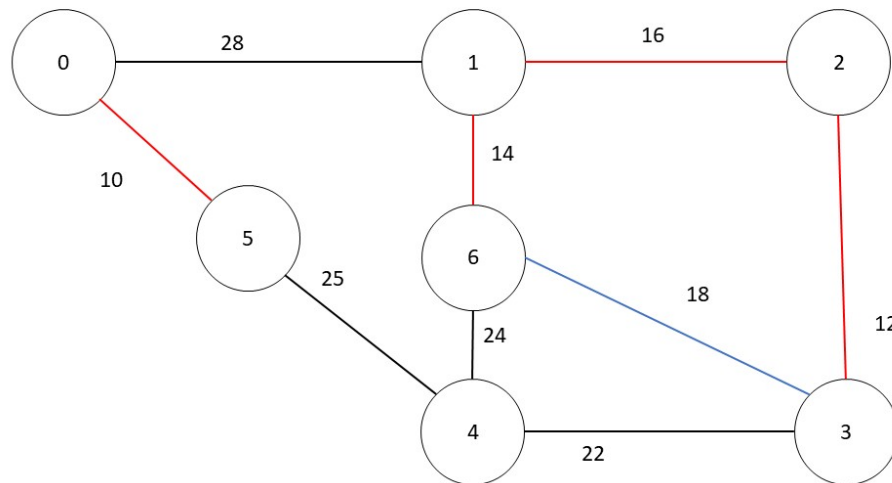
After rejecting the cycle

▼ MST	ArrayList<E> (id=32)
▼ [0]	Edge (id=68)
▪ destination	5
▪ source	0
▪ weight	10
▼ [1]	Edge (id=77)
▪ destination	3
▪ source	2
▪ weight	12
▼ [2]	Edge (id=74)
▪ destination	6
▪ source	1
▪ weight	14
▼ [3]	Edge (id=71)
▪ destination	2
▪ source	1
▪ weight	16
▼ [4]	Edge (id=82)
▪ destination	4
▪ source	3
▪ weight	22
▼ [5]	Edge (id=84)
▪ destination	5
▪ source	4
▪ weight	25

The weight of the last edge was 22. Then we rejected the node with weight 24 because it formed a cycle and we added the node with weight 25 to the MST.



2. Consider the following tree where all red edges denote the edges in an MST and the blue edge denotes the current edge. Set a breakpoint in your find method. Take screenshots of the breakpoint, and values in the parent array. Next, explain the meaning of each value in the parent array and then why a cycle is formed by means of the values in the parent array.



▼ edge	Edge (id=42)
▪ destination	6
▪ source	3
▪ weight	18
⌚ xSet	3
⌚ ySet	3

▼ this	DisjointSet (id=127)
▪ n	7
▼ parent	(id=130)
▶ [0]	0
▶ [1]	1
▶ [2]	2
▶ [3]	3
▶ [4]	4
▶ [5]	5
▶ [6]	6
⌚ i	0
⌚ j	5
⌚ iSet	0
⌚ jSet	5

A cycle is formed

Both vertices lie on the same set (they have the same representative, which is 4 in this case), which means that they are either directly or indirectly connected to each other with the minimum number of edges possible.

▼ MST	ArrayList<E> (id=55)
▼ [0]	Edge (id=44)
▪ destination	5
▪ source	0
▪ weight	10
▼ [1]	Edge (id=57)
▪ destination	3
▪ source	2
▪ weight	12
▼ [2]	Edge (id=58)
▪ destination	6
▪ source	1
▪ weight	14
▼ [3]	Edge (id=59)
▪ destination	2
▪ source	1
▪ weight	16
▪ vertices	7

Current edges on MST before rejecting the Edge with destination 6

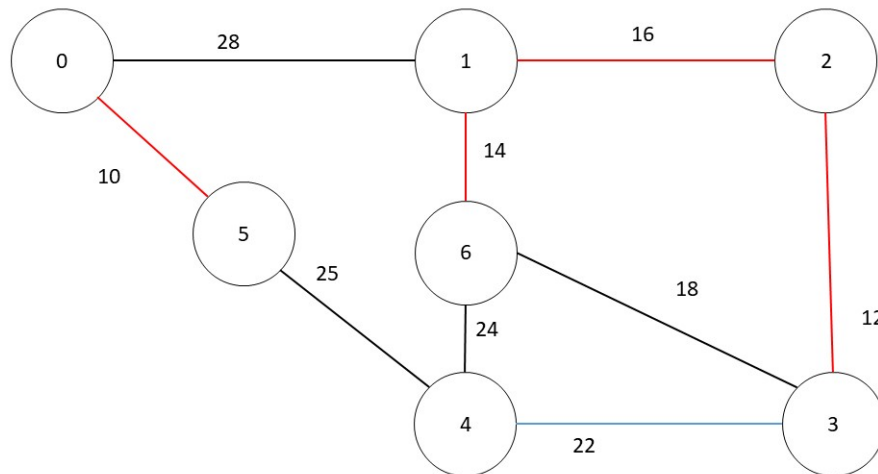
weight 18, source 3 and

Find(x): Return the representative of the set that an element belongs to.

The union-find algorithm helps us to check whether an undirected graph contains a cycle or not. For each edge we make subsets using both vertices of the edge. If both of them are in the same subset, a cycle is formed.

From the image we can see the edge[1] goes from 2 to 3, and the edge [2] goes from 1 to 6. Then, since edge [3] joins 1 and 2, if we were to join 3 and 6, since they have the same representative, we would be making a cycle.

3. Consider the following edge in blue is currently being considered. Set a breakpoint after your find method calls. Take screenshots of the breakpoint, and values in the parent array. Next, explain the meaning of each value in the parent array and then why no cycle is formed by means of the values in the parent array.









```

find() returned
  this
  weight
  pq
  parent
  index
  edge
  edge
    destination
    source
    weight
  xSet
  ySet
  3
  Graph (id=34)
  0
  PriorityQueue<E> (id=36)
  (id=44)
  4
  Edge (id=69)
    4
    3
    22
    3
    4

```

The representative of 4 is `ySet == 4`, and the representative of 3 is `xSet == 3`. Since the representatives are not the same, no cycle is formed. Therefore, we will call `union()` to join the 2 subsets into a single set.

We add the edge [4] to the MST

▼  MST	ArrayList<E> (id=55)
▼  [0]	Edge (id=45)
▪ destination	5
▪ source	0
▪ weight	10
▼  [1]	Edge (id=63)
▪ destination	3
▪ source	2
▪ weight	12
▼  [2]	Edge (id=64)
▪ destination	6
▪ source	1
▪ weight	14
▼  [3]	Edge (id=65)
▪ destination	2
▪ source	1
▪ weight	16
▼  [4]	Edge (id=69)
▪ destination	4
▪ source	3
▪ weight	22