**Prim’s MST Algorithm**

Dr. Byun

12/01/2020

A minimum spanning tree (MST) is a subset of edges that connects all vertices in a graph with the minimum total edge costs. This document describes the operation of Prim's algorithm to get an MST of a graph. If you didn’t watch the following video yet, watch it first to identify the basic idea of the algorithm:

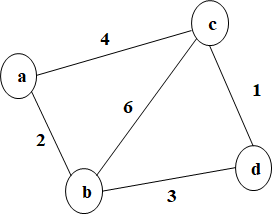
<http://y2u.be/cplfcGZmX7I>

We can summarize Prim's algorithm as follows.

1. Start with a tree T1 consisting of a vertex.
2. Grow T1 by adding one vertex at a time such as T2, T3, …, Tn.
3. Construct Ti+1 from Ti by adding a vertex not in Ti that is closest to those already in Ti. Note that this is **a greedy step** because we select the closest vertex to vertices in Ti.

**Example**

The following is our sample graph, and we will start the Prim’s algorithm from the vertex *a*. Note that the algorithm can start from any vertex of the graph. But in this example, we start from the vertex a.



We can describe this initial configuration of the algorithm (= starting from the vertex *a*) like this:

|  |  |
| --- | --- |
| Tree Vertices | Remaining Vertices |
| a(–, –) | b(a, 2), c(a, 4), d(–, ∞) |

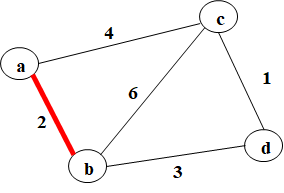
* The left column (= “Tree Vertices”) represents the vertices which are included in the MST so far. At the moment, only the vertex a is included to MST.
* The right column (= “Remaining Vertices”) represents the vertices which are not included to the MST yet.
  + The parenthesized labels of a vertex in the right column indicates the closest vertex to “Tree Vertices” and edge cost.
  + For instance, b(a, 2) means that the vertex b can be reached from the vertex a with the cost 2.
  + Similarly, c(a, 4) indicates that the vertex c can be reached from the vertex a with the cost 4.
  + Since the vertex d can’t be reached from the vertex a, we describe it as d(–, ∞).

From the initial configuration, we should choose a next vertex to be added to MST.

* In the “Remaining Vertices” column, we have three choices such as b(a, 2), c(a, 4), d(–, ∞). Among them, we select the vertex b because it has the minimum cost (= 2).
* Since the vertex b becomes “Tree Vertices”, we should update any values in “Remaining Vertices”, if it’s necessary.
  + This time, the vertex d can be reached from the vertex b with the cost 3. So, we update it to “d(b,3)”.
* We can describe this step like this.

|  |  |
| --- | --- |
| Tree Vertices | Remaining Vertices |
| a(–, –) | b(a, 2), c(a, 4), d(–, ∞) |
| **b(a, 2)** | c(a, 4), **d(b, 3)** |

Visually, we can represent this situation like below.

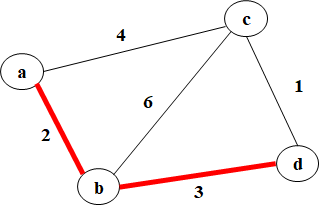


Now, let’s select the next vertex to be added to MST.

* In the “Remaining Vertices” column, we have two choices such as c(a, 4) and d(b, 3). Between them, we should select the vertex d because its cost is 3 which is less than 4 of c(a, 4).
* Since the vertex d becomes “Tree Vertices”, we should update any values in “Remaining Vertices”, if it’s necessary.
  + This time, the vertex c can be reached from the vertex d with the cost 1. So we update it to c(d, 1).
* We can describe this step like this.

|  |  |
| --- | --- |
| Tree Vertices | Remaining Vertices |
| a(–, –) | b(a, 2), c(a, 4), d(–, ∞) |
| b(a, 2) | c(a, 4), d(b, 3) |
| **d(b, 3)** | **c(d, 1)** |

Visually, we can represent this situation like below.

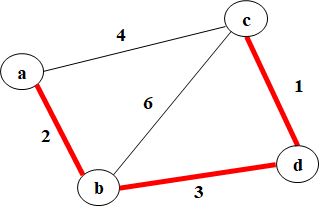


Finally, let’s select our last vertex to be added to MST.

* Since the “Remaining Vertices” column has only one choice, c(d, 1), we select it and move it to “Tree Vertices”.
* We can describe this step like this.

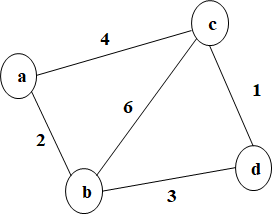
|  |  |
| --- | --- |
| Tree Vertices | Remaining Vertices |
| a(–, –) | b(a, 2), c(a, 4), d(–, ∞) |
| b(a, 2) | c(a, 4), d(b, 3) |
| d(b, 3) | c(d, 1) |
| **c(d, 1)** |  |

Visually, we can represent this situation like below.



**Exercise 1**

In the previous example, we conducted the Prim’s algorithm from the vertex *a*. In this exercise, start the algorithm from the vertex c.



In other words, fill out the following table. Note that the first “Tree Vertices” is “c(–, –)” and you have to provide the values for the vertices a, b, and d in “Remaining Vertices”. After that, fill out the 2nd, 3rd, and 4th rows.

|  |  |
| --- | --- |
| Tree Vertices | Remaining Vertices |
| c(–, –) | a( , ), b( , ), d( , ) |
|  |  |
|  |  |
|  |  |

**Do not see the answer immediately.** Try to solve it by yourself.

This is the [sample solution](https://docs.google.com/document/d/15Yk-Mc-niJoUi1QrIt1Xe6DLQvwxsvGgCe8YIOwUpcA/edit).