# HANDOUT 09: MINIMUM SPANNING TREES

This handout could help to prepare for the Written Assignment 09.

## 1.1 Prim's Algorithm

(Goodrich2011, p.651) defines Prim's algorithm. It finds a minimum spanning tree in an undirected graph with given edge weights. See also https://bit.ly/2VLz3DK. It is an efficient algorithm; it requires $O((m+n)\log_2 n)$ time, if we use priority queues. In this exercise you do not need to implement a priority queue; assume that you can always compute the minimums in your head and grow the MST accordingly.

### 1.1.1 Problem

We start with the graph shown in Figure:

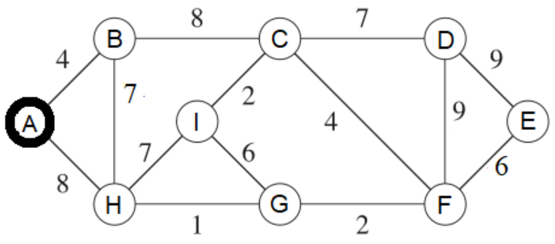

Fig. 1: Graph Diagram for Prim's Algorithm.

**(A)** Vertex $A$ will be your source vertex. It is the first vertex added to the MST vertice set $S$. At every step you find the lightest edge that connects some vertex in $S$ to some vertex not in $S$. Add this new vertex to a graph and remember the edge you added. Show how the Prim's MST (Minimum Spanning Tree grows) one edge at a time.

**Note:** In cases when there is a choice between multiple lightest edges of the same weight, pick the edge $(v,w)$ with $v \in S$ and $w \notin S$ such that $(v,w)$ lexicographically precedes any other lightest edge.

**(B)** Redraw the graph, highlight the edges selected for MST (make them bold or color them differently). Add up the total weight of the obtained MST and write this in your answer (it should be the minimum value among all the possible spanning trees in this graph).

### 1.1.2 Solution

**(A)** At each step we show the current set of vertices in MST (denoted by $S$) and which edge is being added.

1. $S = \{A\}$, adding edge $AB$
2. $S = \{A, B\}$, adding edge $BH$
3. $S = \{A, B, H\}$, adding edge $HG$
4. $S = \{A, B, G, H\}$, adding edge $GF$
5. $S = \{A, B, F, G, H\}$, adding edge $FC$
6. $S = \{A, B, C, F, G, H\}$, adding edge $CI$
7. $S = \{A, B, C, F, G, H, I\}$, adding edge $FE$
8. $S = \{A, B, C, E, F, G, H, I\}$, adding edge $CD$

**(B)** Solution shows the MST edges added in previous step colored blue:

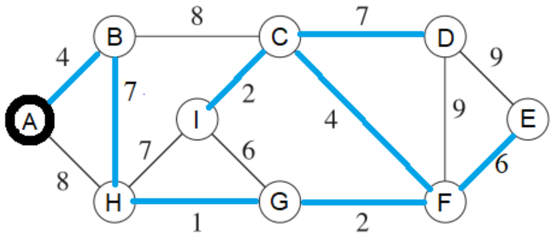

Fig. 2: MST obtained by Prim's Algorithm.

The total weight of this MST is $4 + 7 + 1 + 2 + 4 + 2 + 6 + 7 = 33$. (*In this case the MST is unique. In general case there is no guarantee that there are no other MSTs of the same weight, but the one we found with Prim's algorithm is among the lightest ones.*)

## 1.2 Kruskal's Algorithm

### 1.2.1 Problem

**(A)** Run Kruskal's algorithm on the same graph as in the previous subsection (*Prim's algorithm*). After each step when there is an edge connecting two sets of vertices, write that edge and show the partition where that edge connects two previously disjoined pieces in the forest of trees.

**Note:**

**If there are multiple lightest edges that can be used to connect two disjoined pieces, pick edge** $(v, w)$ which lexicographically precedes any other.

**(B)** Redraw the given graph (show the order how you added the edges in parentheses). Also compute the total weight of this MST.

### 1.2.2 Solution

**(A)** We list the steps that add edges and join two previously disconnected pieces:

1. Add edge $GH$, the partition becomes $\{A, B, C, D, E, F, GH, I\}$.
2. Add edge $CI$, the partition becomes $\{A, B, CI, D, E, F, GH\}$.
3. Add edge $FG$, the partition becomes $\{A, B, CI, D, E, FGH\}$.
4. Add edge $AB$, the partition becomes $\{AB, CI, D, E, FGH\}$.
5. Add edge $CF$, the partition becomes $\{AB, CFGHI, D, E\}$.
6. Add edge $FE$, the partition becomes $\{AB, CEFGHI, D\}$.
7. Add edge $BH$, the partition becomes $\{ABCEFGHI, D\}$.
8. Add edge $CD$, the partition becomes $\{ABCEFGHID\}$.

**(B)** Solution shows the MST edges added in previous step colored blue. The total weight is 33. The order of their addition is shown in red in parentheses.

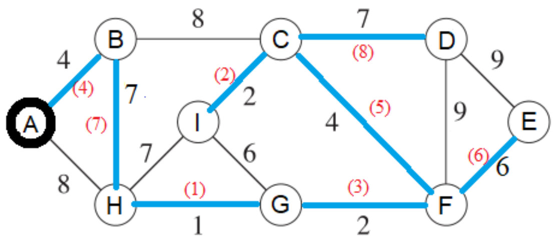

Fig. 3: MST obtained by Kruskal's Algorithm.

**Note:** In some cases Prim's and Kruskal's algorithm can yield different MSTs even for the same input graph, but they are both optimal in such cases.