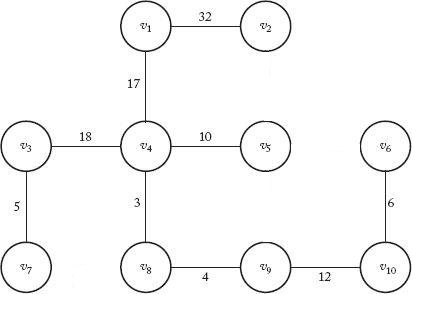
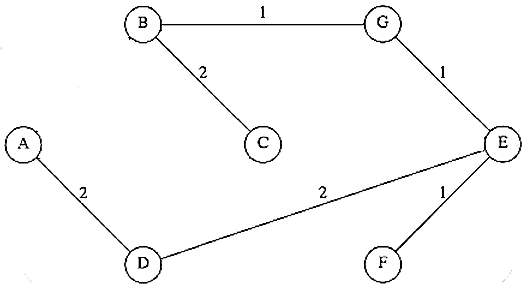
1. F = {(v1, v4), (v4, v8), (v8, v9), (v4, v5), (v9, v10), (v10, v6), (v4, v3), (v3, v7), (v1, v2)}

32 + 17 + 18 + 5 + 3 + 10 + 4 + 12 + 6 = 107

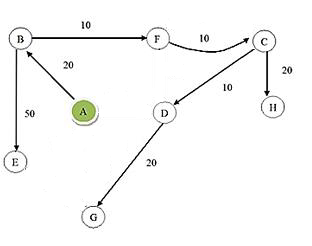


1. F = {(B, G), (G, E), (E, F), (E, D), (D, A), (B, C)}

1 + 1 + 1 + 2 + 2 + 2 = 9



|  |
| --- |
| 1. Prim's algorithm is  Ө(n^2), and Kruskal's algorithm is  Ө(m∙lgm), where m is the number of edges in the graph. |
| Since our graph is completely connected, m = n(n-1)/2. Kruskal's algorithm in terms of n is therefore Ө((n^2)∙lgn), slightly less efficient than Prim's. |



1. Dijkstra’s algorithm finds the minimum distance over all of nonnegative values from starting point

Prim’s algorithm finds the minimum spanning tree for a given graph

Therefore, Prim’s algorithm stores the minimum cost edge whereas Dijkstra’s algorithm stores the total cost from the source to the destination vertex.