Borůvka's algorithm

Evan Alba

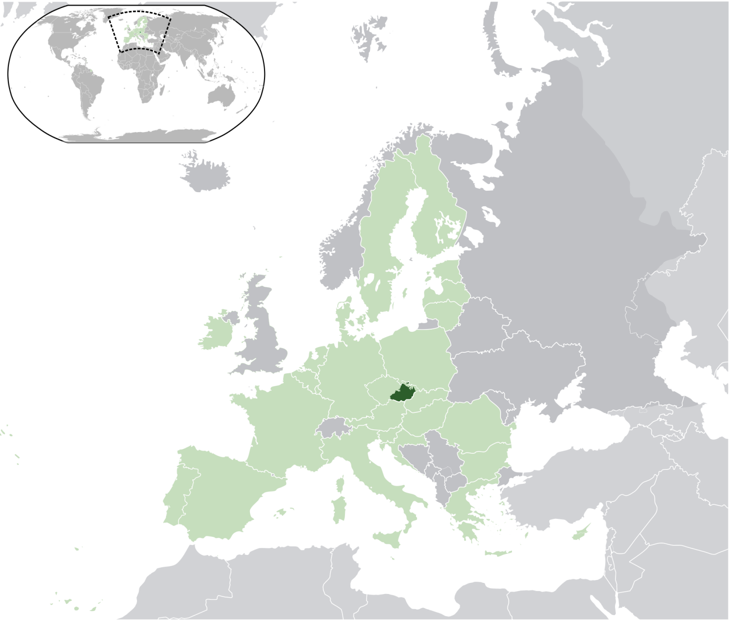
Introduction

- Borůvka's algorithm is a greedy algorithm for finding a minimum spanning tree in a graph.

Introduction

- Borůvka's algorithm is a greedy algorithm for finding a minimum spanning tree in a graph.
- It was published in 1926 by Otakar Borůvka, a Czech mathematician as a method of constructing an efficient electricity network for the South-Moravia district in the region Moravia. (Fun Fact: The first textbook on graph theory was written by Dénes Kőnig, and published in 1936.)

Moravia



Otakar Borůvka

Applications of Borůvka's algorithm

- A method of constructing an efficient electricity network for the South-Moravia district.

Applications of Borůvka's algorithm

- A method of constructing an efficient electricity network for the South-Moravia district.
- Especially in the parallel computing literature, Borůvka's algorithm is frequently called **Sollin's algorithm**. (Named after the computer scientist George Sollin.)

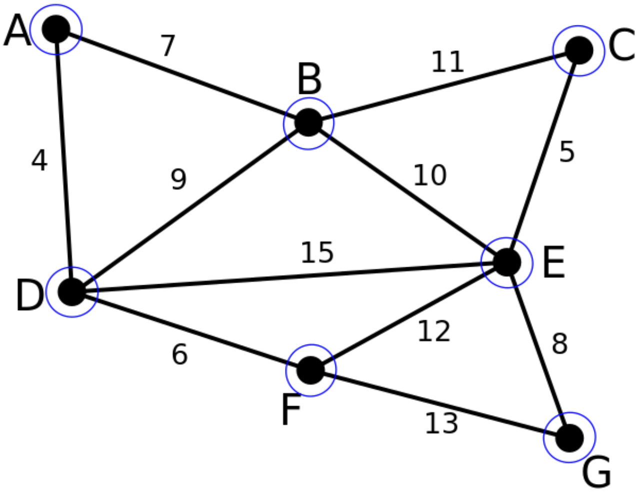
MST Problem

Given a connected (undirected) graph $G = (V, E)$ with real weights assigned to its edges. Find a spanning tree (V, T) of $G(i.e. T \subseteq E)$ with the minimal weight $w(T)$.

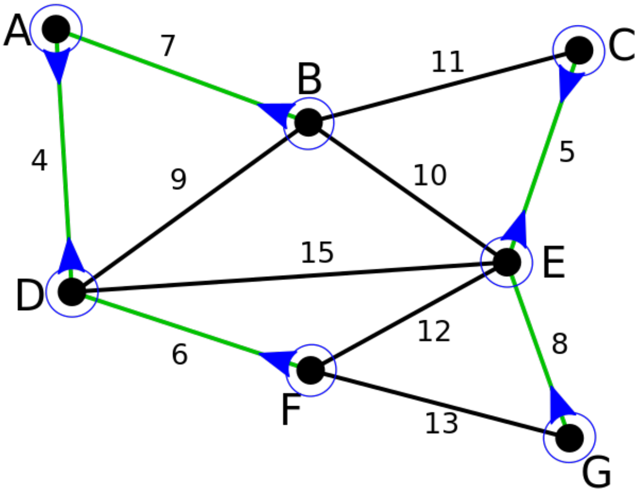
Intuition

1. Start with isolated vertices. (Initially, each vertex in the graph is considered an isolated component.)
2. Find the cheapest edges. (During each iteration, we find the cheapest edge that connects two currently separate components. If we have a tie in the cheapest edge, we pick the cheapest edge we have seen first.)
3. Add the cheapest edges to the minimum spanning tree. (Note: We add the cheapest edges as long as they do not cycles.)
4. Repeat until all are connected. (We repeat the process of finding cheapest edges and add them to the minimum spanning tree until all vertices are connected in a single minimum spanning tree.)

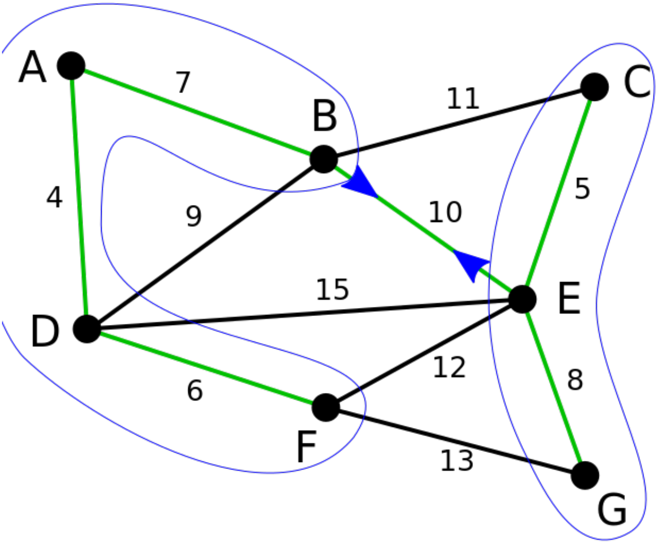
Example



Example



Example



Pseudocode

```
function BORUVKA(G):
    INITIALIZE-SINGLE-COMPONENT(G)
    // Let A = Minimum Spanning Tree Set
    A = ∅
    while A does not form a spanning tree
        for each tree T in G.forest
            e = MINIMUM-WEIGHT-EDGE(T, G)
            A = A ∪ {e}
            if FIND-SET(e.u) ≠ FIND-SET(e.v)
                UNION(e.u, e.v)
    return A
```

Runtime

- The runtime for Borůvka's algorithm is $O(|E|log|V|)$.

Runtime

- The runtime for Borůvka's algorithm is $O(|E| log |V|)$.
- A faster randomized minimum spanning tree algorithm based on Borůvka's algorithm due to Karger, Klein, and Tarjan runs in expected O(E) time.

Runtime

- The runtime for Borůvka's algorithm is $O(|E| log |V|)$.
- A faster randomized minimum spanning tree algorithm based on Borůvka's algorithm due to Karger, Klein, and Tarjan runs in expected O(E) time.
- The best known (deterministic) minimum spanning tree algorithm by Bernard Chazelle is also based on Borůvka's algorithm and runs in O(E α(E,V)) time, where α is the inverse Ackermann function (a very slowly growing function.).