

Borukva's Algorithm

November 16, 2022

0.1 Principles Of Boruvka's Algorithm

Boruvka's algorithm is a greedy algorithm to find the minimum spanning tree (MST) of a given graph. It does this by first finding the minimum cost edge for each node in the graph, if two nodes share the same minimum cost edge they are connected into their tree. Once you have done that for all nodes in your graph you will have many smaller trees. Then, for all your trees that you have just made you find the minimum cost edge that is connected to another one of your trees, creating a smaller amount of larger trees. You repeat this step until you have only one tree remaining. The one remaining tree will be your MST.

The main advantage of Boruvka's algorithm over something like Kruskal's algorithm is that it can be parallelised easily, this is because each node finds its own minimum cost edge independently of all other nodes because they do not consider where the edge is going, only that it is the minimum. With this knowledge we can give each thread a node and all its connected edges and compute the minimum cost edge of each node simultaneously. This makes the algorithm very scaleable and good for large graphs.

0.2 Pseudo Code

Boruvka's Algorithm

Inputs : A graph G with set of vertices V and edges E

Output : The minimum spanning tree of graph G

$V_T \leftarrow V$

$E_T \leftarrow \emptyset$

while $|V_T| > 1$ **do**

for $T \in V_T$ **do**

$M_n \leftarrow \emptyset$

for $N \in T$ **do**

$M_n \leftarrow \text{append}(\text{minCostEdge}(N))$

end for

$\text{append}(E_T, \text{min}(M_n))$

$V_T \leftarrow \text{combineTrees}(V_T, E_T)$

end for

end while

return V_T, E_T