CS-E3190 Principles of Algorithmic Techniques

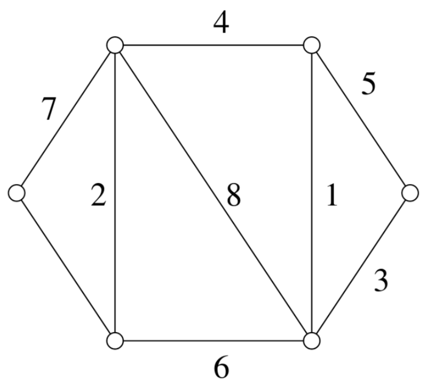
04. Local search – Tutorial Exercise

1. **Borůvka's algorithm.** The goal of this greedy algorithm is to compute a minimum spanning tree in a weighted undirected graph G. Let F be the set that will contain the edges of the minimum spanning tree. The algorithm works by contracting edges i.e. turning an edge and two nodes into a single node. Even if the graph G is simple (i.e. no loops or multiple edges), contractions might give rise to multiple edges between a same pair of nodes or to loops i.e. edges that "start and end" at the same node. Hence, replacing multiple edges by a single one and removing loops will occur in the algorithm.

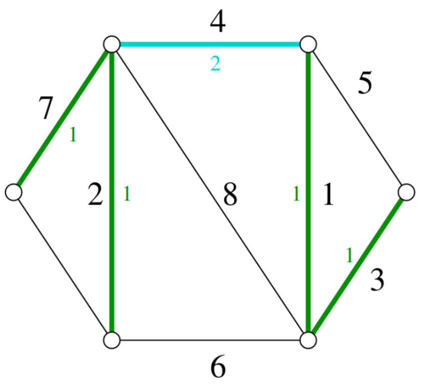
Algorithm 1: Borůvka's algorithm

```
F = ∅
while G is not reduced to a single node do
    Remove loops from G
    Replace multiple edges by a single edge, weighted by the
      minimum of the weights from the multiple edges
    for x node of G do
        Find the edge e_x of minimum weight adjacent to x
        F = F ∪ {e_x}
        Contract e_x
    end
end
return F
```

(a) Apply the algorithm to the following graph:



Solution.



The coloured numbers are the phase numbers the edges were picked in. In this example, there are only $2 = \lfloor \log_2 6 \rfloor$ phases.

(b) Prove that this algorithm is correct and terminates **Solution.** We first need to prove the following statement: $\forall v \in V(G)$, the minimum spanning tree of G must contain the minimum weight edge incident on v. Then, notice that at each contraction, the algorithm creates a forest in the original graph. This forest does not have any edges that would not be in the MST. Finally, the algorithm runs until there is one node left in G. This implies that the edges in F must correspond to the MST's edges. Let us now prove this statement by contradiction:

Suppose the MST does not contain the minimum weight edge adjacent to the node x. Then, there must be another edge adjacent to x in the MST. However, if this edge is replaced by the minimum weight edge, the total weight of the new tree would be less than or equal to the weight of the MST we considered in the first place. This is a contradiction since the MST should have had minimum weight.

(c) Prove that this algorithm runs in $O(m \cdot \log n)$ time.

Solution. Each time we go through the while loop of the algorithm, the number of nodes left in G is divided by two. Indeed, for each node an edge is contracted, which "turns two nodes into one". Hence, if n is the number of nodes in G, the while loop runs $\log n$ times. Inside the while loop, we go through all the edges that are left in G. If there are m edges in G at first, going through the while loop once takes at most m steps. Hence, the time complexity of this algorithms is $O(m \cdot \log n)$.

(d) Prove that the time complexity of this algorithm is linear for planar graphs.

Solution. Each phase of the Borůvka algorithm takes $O(n)$ time. In simple planar graphs, $m \leq 3n$, planar graphs remains planar after edge contractions/deletions, and the number of nodes (at least) halves in each phase. Hence, the overall running time is less than pr equal to $cn + cn/2 + cn/4 + cn/8 + \cdots = O(n)$.