

MTH6412B Project - Report of Phase 3

Authors:

- Luis Rojo-González

Description: You may find the scripts used in this notebook in the following [GitHub repository](#).

Procedure

The adjacency matrix has been added to the **build_graph** function in order to use it as input for Prim's algorithm. In here, this implementation considers a starting node which might be given by the user; otherwise, the script makes a randomly select a node to start the procedure.

A new version of Kruskal's algorithm (**kruskal_V2.jl**) has been created to include both heuristics following the i) **path compression** and ii) **union by rank** guidelines. The implementation considers that the path compression is used as an input for the union by rank procedure. Besides, it is worth noting that **path_compression** function is an extension of the **trace_parent** function presented in the previous Kruskal's algorithm version.

Question

One should notice that the rank of a tree is the number of edges that connect the nodes in the graph. Thus, if we consider a graph $G = (N, E)$, where N is the set of nodes and E is the set of edges; then, since there has not to be cycles, the number of edges in it is $|N| - 1$.

Then, following the dive-and-conquer scheme involved in the union by rank technique, one has two subproblems of size $i - 1$ where i is the rank of the tree. Despite this, the relation that exists corresponds to an inequality since the rank of a tree might be either the same or greater after 'merging' two trees. Hence, we have that $N(i) \geq 2N(i - 1)$, where $N(i)$ represents the minimal number of vertices of a tree having a root of rank i . Thus, by recursion we have that $N(i) \geq 2^i N(0)$.

Therefore, due to $N(i) \leq |S|$ has always to be met, we have that $2^i \leq |S|$ that implies that $i \leq \log_2(|S|)$.

Moreover, as we consider that the number of nodes in a tree has to be an integer number, then one has that $i \leq \lfloor \log_2(|S|) \rfloor$.

Running times

As one may see in the plot below, the running times increases in function of the number of nodes. Besides, it is clearly see that the use of the heuristics under study improve the performance of the algorithm. On the other hand, according to results obtained by Prim's algorithm, one should consider its implementation avoiding the use of adjacency matrix to speed it up.

Here, the plot shows the running time for the previous version of Kruskal's algorithm (Kruskal, in red), its updated version, i.e., including the heuristics, (Kruskal_2, in blue) and Prim's algorithm (Prim, in green).

