Pedibus Challenge Documentation

Gabriele Bressan, Pietro Di Marco, Simone de Santis

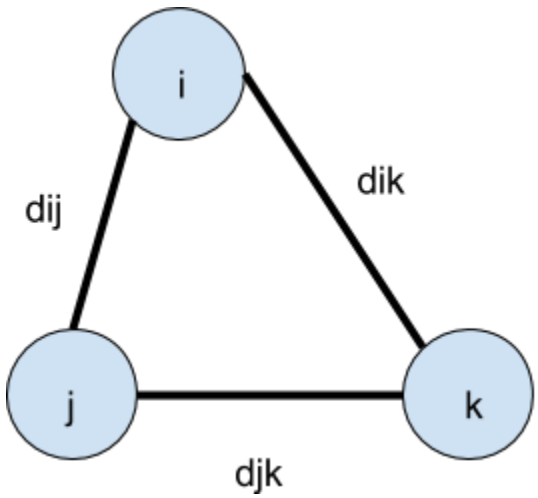
For this challenge we focused on three principal goals write in order of importance that are:

1. Minimization of nodes number
2. Minimization of dangerous value about chosen path
3. Execution time of the algorithm

How we solved these goals

Minimization of nodes number

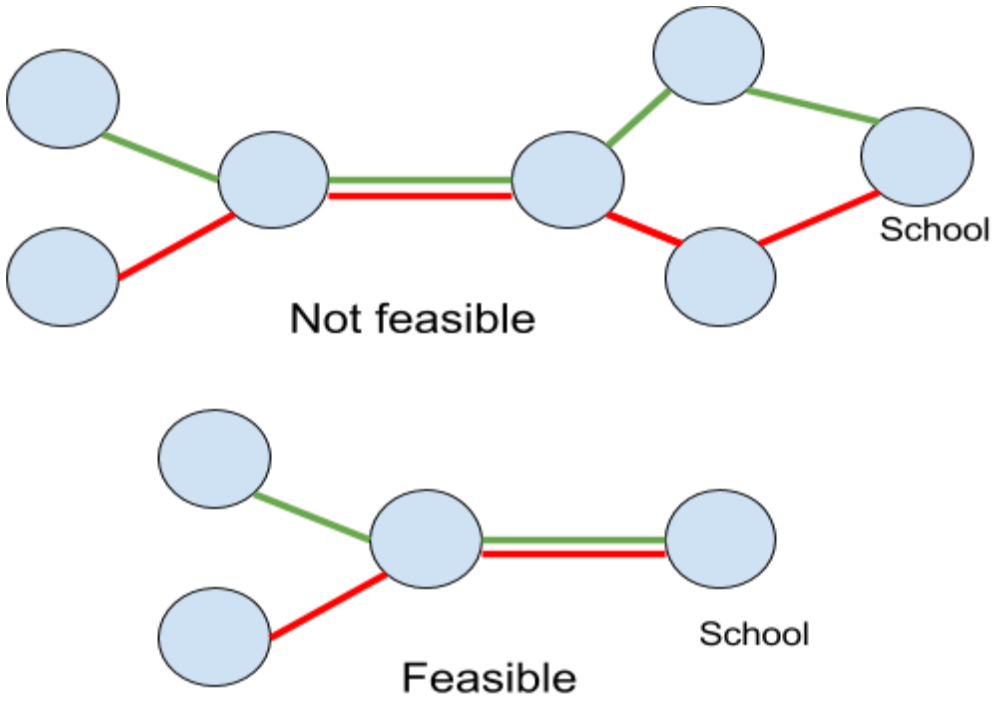
For this part we decide to use an algorithm that start to analyze the problem from the school (node zero) to the each leaf.
Our algorithm start from the node zero (school) and try each possible path that respect the condition given on the exercise text:



so the condition is dij+djk>ALPHA*dik

Once that the algorithm reached the **leaf**, come back to the node zero (school) and the node zero will decide what is the path with the most number of nodes that can be reached from it and we put these paths into an ArrayList.

After that on the ArrayList that contains evaluated paths, each time, algorithm take the path that cover more number of not connected nodes and we eliminate from the ArrayList all paths that intersect with the path taken and in this way we respect the rule given during the description of the problem (if two paths merge between them, they cannot divide again), this help us to decrease number of comparison in future steps. We repeat this process until we cover all nodes of the given graph.



Definition of leaf: Leaf is a node that cannot reach any other node because doesn't respect the condition of triangle inequality.

Minimization of dangerous value about chosen path

The algorithm that we applied to minimize the dangerous in the graph works by choosing every time, from paths with the same number of node, that one containing less danger.
When the choice has been performed, the dangers of that specific path, in the dangerous matrix, are setted to 0, in this way in the following iterations a paths that follows a path already chosen has the precedence among the others.

Execution time of the algorithm

Since there is not time to explore all the possible path in graph from the school to the leaf we decided to introduce a value called TRESHOLD in our code.
TRESHOLD is a parameter that is proportional with the number of nodes of the graph given and **alpha** value, and permit to our code to terminate until 1 hour from the start of program execution. In this way we

found a feasible solutions related to the value of this parameter. For this delivery we chosen a better compromise between time and solution.