**ReflexActInt2_A01638996**

To resolve the four problems that were given to us, we implemented Kruskal, Nearest Neighbor (TSP), Ford Fulkerson, and Voronoi algorithms, each of these algorithms allowed us to solve in an efficient manner all the problems.

For the first program we used Kruskal algorithm to determine the optimal way to connect the neighborhoods with optic fiber, this algorithm allowed us to create a minimum spanning tree (MST). This algorithm sorts every edge by its weight (this case being distance) and while iterating it begins to join them (as long as they do not create cycles and as long as the vertexes are not part of the same union already). Thanks to this we could obtain the way we can join all neighborhoods with the least cost possible. It´s complexity is $\boldsymbol{O(E \log(E))}$ , the most efficient way to resolve this type of problem (to the extense of our knowledge).

For the path that the employes, that are delivering physical goods, we decided to use the algorithm of traveling salesman problem (nearest neighbor), we know its limitation of being an approximation, however we opted with this approach, given that the other methods were to complicated and to time expensive (for us) to complete. This algorithm starts in a neighborhood and connects to the closest neighborhood that has not been previously visited, the algorithm repeats this until it reaches yet again our first neighborhood. This algorithm offered us a simple implementation and an ok complexity, $\boldsymbol{O(N^2)}$ , for the problem.

In thew third problem, to calculate the maximum flow of information between two nodes, we decided to implement the Ford-Fulkerson method using BFS, that works by making an augmentation path and a flow and residual network to try to find this path. This algorithm ensured us the maximum flow of information between the nodes. It has a complexity of $\boldsymbol{O(VE^2)}$, the best complexity that we could have archived.

Finally, to assign an area of control for the “exchanges”, we decided to calculate the Voronoi diagram utilizing the library CGAL (we decided on this for two reasons our lack of knowledge for the algorithm and because the professor allow us to). This algorithm divides coordinates in two parts recursively and then joins the lines to generate defined areas of control. It´s complexity is $\boldsymbol{O(N\log(N))}$, this is the best algorithm (in our knowledge) to use in this occasion.

As a final note, I can say we archived an optimal (not the best) solution for the integral activity as a whole, that combined the knowledge of graphs that we´ve been acquiring through the second part of this semester.