Activity 1. Prim Algorithm

My version of the Prim algorithm has complexity O(n^3), there may be an approach that reaches O(n^2) but I couldn’t get it, it was specially possible as the matrix provided has bidirectional edges and half of it isn’t needed to be iterated.

Results obtained

|  |  |  |
| --- | --- | --- |
| n | t Prim(s) | tPrim(ms) |
| 32 | 0.013 | 13 |
| 64 | 0.117 | 117 |
| 128 | 1.637 | 1637 |
| 256 | 13.621 | 13621 |
| 512 | 222.181 | 222181 |
| 1024 | OOT | OOT |
| 2048 | OOT | OOT |
| 4096 | OOT | OOT |
| 8192 | OOT | OOT |

As The complexity is o(n^3) it makes sense that it goes OOT so fast, I had 1024 running for almost 20 minutes and it still didn’t give the solution. In conclusion, this is a very poorly optimal approach, specially for bigger matrixes, for the smaller ones it works just fine.