Discrete Optimization
Task 1 - Shortest path

Student: Horace Guy Report on Task 1, MA2827 2017

1 Work division

I worked in collaboration with Olivier on this task. We looked at the test files, and found that the 3 first graphs are positive-weighted, the 4th one is not, and the fifth only has weights with value 2.

Thus, in regard of the algorithms complexity we saw in class, we decided to implement each algorithm : Dijkstra for the first 3, Bellman-ford or Floyd-warshall for the 4th, checking whichever is the fastest, and breadth-first search for the last one.

We split the work between Olivier, he coded the glouton-approaches of Dijkstra and BFS whereas I developed the negative-weights algorithms Bellman-Ford (BF) and Floyd-Warshall (FW).

2 Strategy adopted

I used the `numpy` module for the distances matrix and the result array. For the BF algorithm, I sorted the pairs of nodes by order of the source point `s`, in order the avoid a re-computation of the distances if a same source matches multiple targets in the test file.

As BF runs in $\mathcal{O}(|A||V|)$ where A are the arcs and V the vertices, but only compute the distances for a single source, we have to iterating it `len(node_pairs)` times, which we will denote $|T|$ (for test sample). The total complexity is then $\mathcal{O}(|A||V||T|)$ for BF.

On the contrary, FW runs in $\mathcal{O}(|V|^3)$ *but* computes every distances, such that one does't need to iterate. For a dense graph, the second method is of course more efficient (because $|A| = \frac{n(n-1)}{2}$

In the example 4, every vertes has 4 neighbors. Thus, there are $4|V|$ arcs. The test sample is of magnitude 100, which is lower than $\frac{|V|}{100}$. Thus, the BF method is more efficient in this case, that's why I decided to use it in the final program.

As a result, the program completed the test 4 in approximately 4 minutes on my machine, which is quite slow. It is probably due to the use of the zip function, which could have been avoided by directly taking the 4 neighbors of each node.

The FW implementation didn't finish in less than 10 minutes.