

ALL-PAIRS SHORTEST PATH PROBLEM

FRANCESCO CAPORALI

MAY 23, 2023

ABSTRACT

Graph theory plays a central role in many areas: numerous real-world problems and phenomena have a natural abstraction in graphs. One of the most discussed of these is the problem of connecting different routers in order to distribute information in a computer network. Specifically, we are going to analyse possible routing algorithms in link-state protocols, i.e. in which the topology of the entire network and all link costs are known to the routers. In the following project, the problem in question will be transposed in the form of the shortest path problem. In this context, various solving approaches will be analysed, both on real data and on randomly generated data with a structure similar to the one considered. Our final goal is to explore several possible solutions to the subproblem conventionally referred to as the *all-pairs shortest path problem*, comparing their costs in space and time. Various algorithms implemented using different data structures introduced during the course, such as queues, priority queues and lists, are discussed. The aim is also to show how the cost of an algorithm can be reduced by changing only the data structure with which it is developed.

1. MAIN

If $G := (V, E)$ is our graph (free of negative cycles) with $n = |V|$ and $m = |E|$, it holds:

- Floyd-Warshall
 - *weights*: positive/negative;
 - *structures*: 2-dimensional array;
 - *time complexity*: $O(n^3)$ (actually $\Theta(n^3)$);
 - *space complexity*: $O(n^2)$ (actually $\Theta(n^2)$);
 - *link*: wikipedia.
- Dijkstra:
 - *weights*: positive;
 - *structures*: priority queue (Fibonacci heap);
 - *time complexity*: $O(n^2 \log(n) + nm)$;
 - *space complexity*: $O(n^2)$ (is $nO(n) = O(n^2)$);
 - *link*: wikipedia (space complexity opengenius).
- Johnson:

- *weights*: positive/negative;
- *structures*: priority queue (Fibonacci heap) for Dijkstra / 1-dimensional array for Bellman-Ford;
- *time complexity*: $O(n^2 \log(n) + nm)$;
- *space complexity*: $O(n^2)$;
- *link*: [wikipedia](#) (space complexity [geeksforgeeks](#)).