

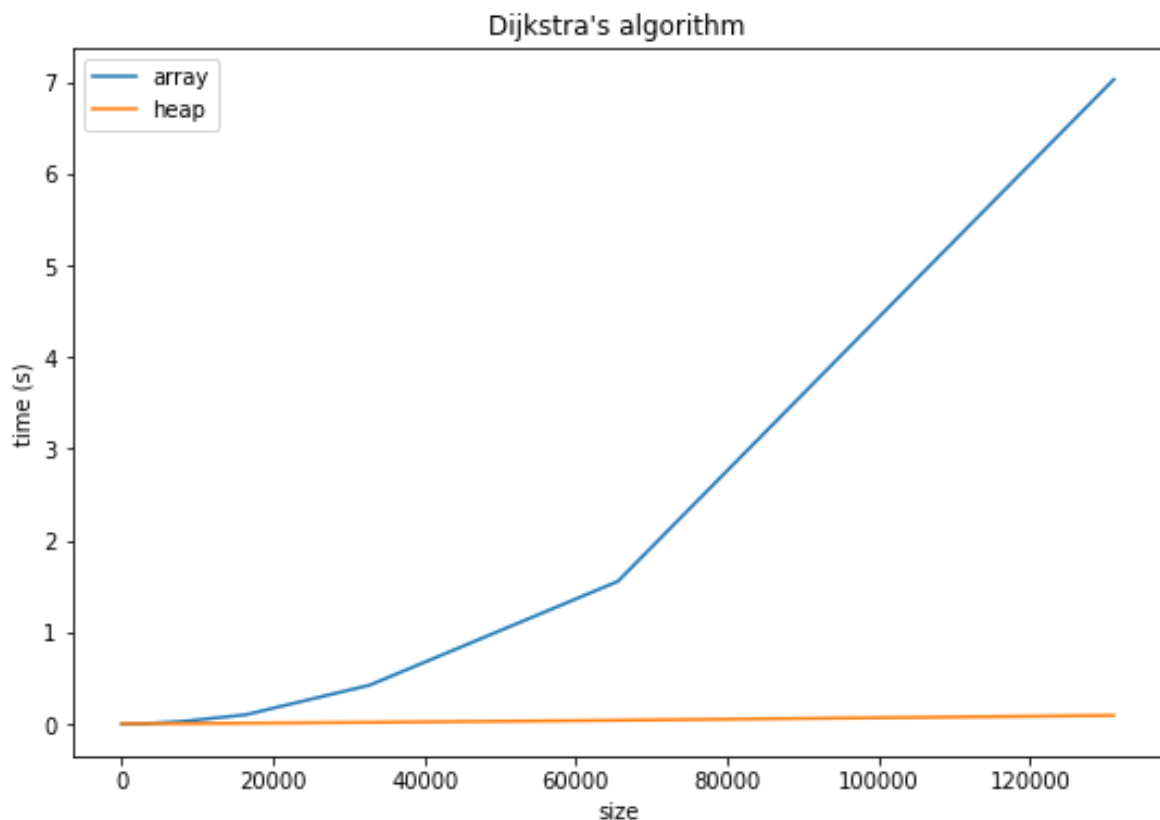
# Dijkstra's algorithm

This folder contains my implementation of Dijkstra's algorithm. I compared two ways of implementing this algorithm: array based and heap based.

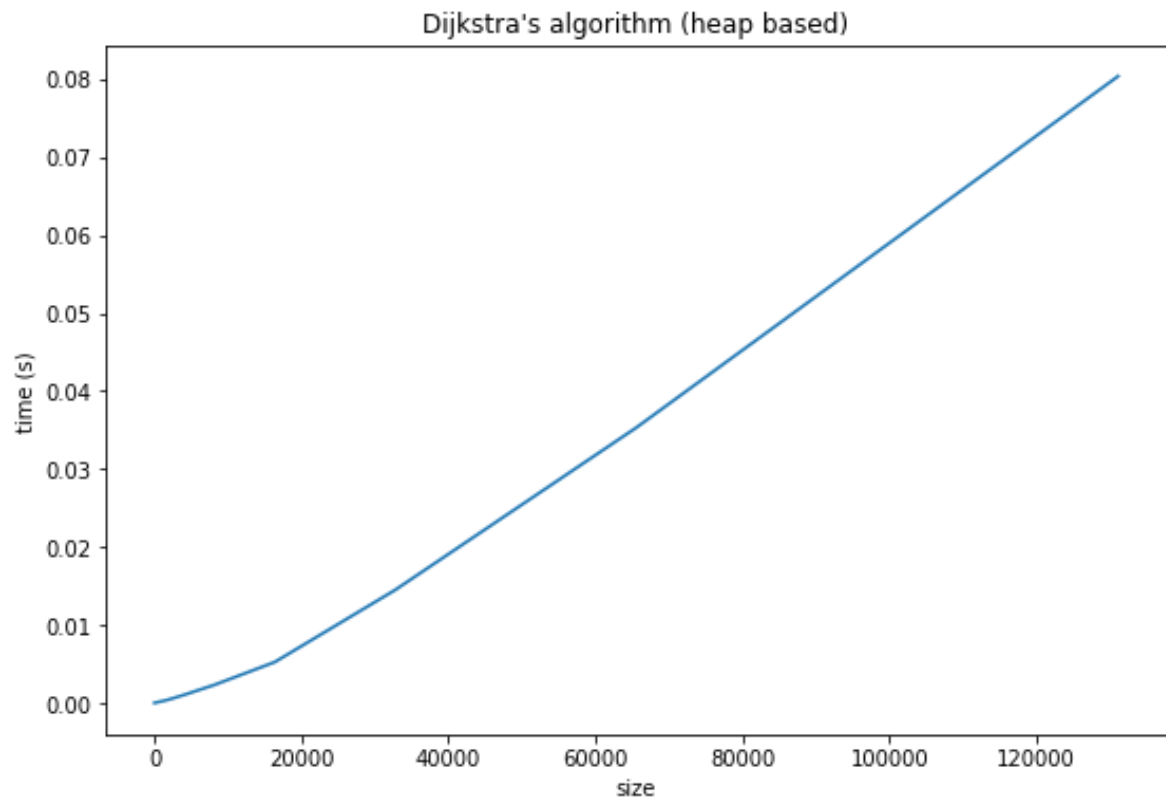
The theoretical complexity of Dijkstra's algorithm is  $\Theta(|V|^2 + |E|)$  for the array based version and  $O((|V| + |E|)\log(|V|))$  for the heap based one, where  $|E|$  is the number of edges and  $|V|$  is the number of nodes. In `main.c` both versions of Dijkstra's algorithm are tested on randomly initialized graphs of different sizes, each node having an average number of exiting edges of 5.

Then, on average, the number of edges  $E$  is 5 times the number of nodes  $V$ , meaning that approximate formulas for complexity are  $\Theta(|V|^2)$  for array based and  $O(|V|\log(|V|))$  for heap based algorithm.

The following figure shows the results in terms of time of running both versions on different graph sizes (results are averages of 10 runs on the same size):



As expected, the array based version is definitely much slower than the heap based one, whose execution time is negligible if compared to the one of the array based one. To have a better idea of the behaviour of the heap based algorithm, next figure shows the curve of times for this version alone:



The plots confirm the theoretical formulas obtained for complexity, showing a quadratic growth on the array based version of Dijkstra's algorithm and a slightly more than linear growth on the heap based one.