06-Report

July 13, 2020

1 Weighted Graphs: Homework

1.1 Exercise 1

- Implement the array-based version of the Dijkstra's algorithm.

Solution:

Implementation of array-based version of the Dijkstra's algorithm can be found in `src/dijkstra.c` file. For this and other version, data types that is used in program are defined such as `Node`, `Array` and `graph`. Implementation is based on the pseudo-code that is provided in lecture notes.

1.2 Exercise 2

- Implement the binary heap-based version of the Dijkstra's algorithm by using the library binheap that was developed during Lesson 6, Lesson 7, and Lesson 8.

Solution

Implementation of array-based version of the Dijkstra's algorithm can be found in `src/dijkstra.c` file. `Binary heap` and its functions were already implemented during the lectures so this implementation is used in the program. It is based on the pseudo-code that is provided in lecture notes.

1.3 Exercise 3

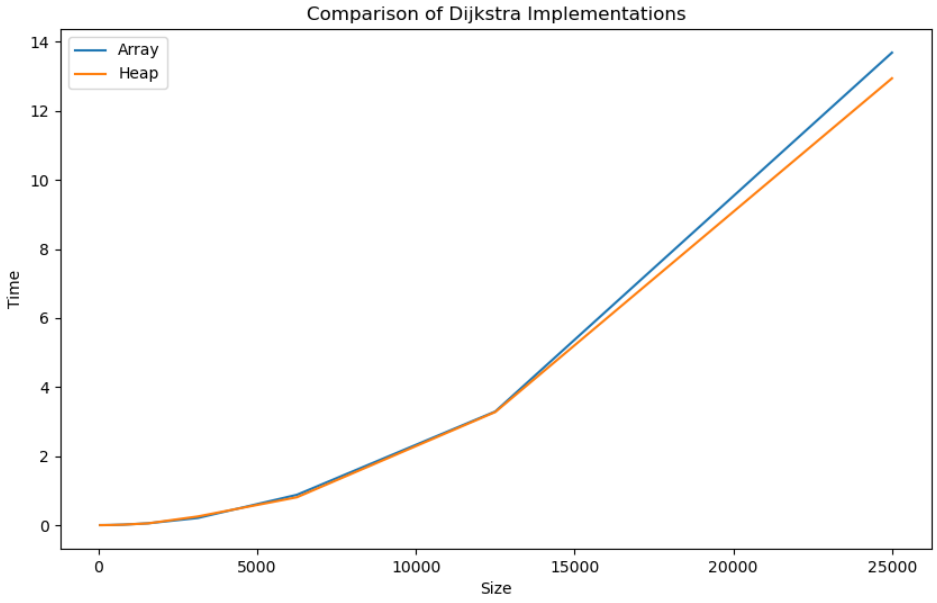
- Test the implementations on a set of instances of the problem and compare their execution times.

Solution

For testing correctness and performance of implementation `src/main.c` is written. Below you can find the theoretical complexity of array-based and binary heap-based dijkstra algorithms as well as plot that is produces by performance test results.

Algorithm	Time Complexity			
	Build	**Extract**	**Update Distance**	**Complexity**

Algorithm	Time Complexity			
Array-based	$\Theta(n)$	$\Theta(n)$	$\Theta(1)$	$\Theta((\|\|V\|\|)^2 + \|\|E\|\|)$
Heap-Based	$\Theta(n)$	$\mathcal{O}(\log n)$	$\mathcal{O}(\log n)$	$\mathcal{O}((\|\|V\|\| + \|\|E\|\|) * \log \|\|V\|\|)$



It seems that plot shows that heap-based implementation shows better performance but it is not as significant as theoretical complexity. One reason could be inefficient implementation of `relax`.