Problem session 3

Dibran Dokter 1047390 & Marnix Lukasse 1047400

September 26, 2019

4

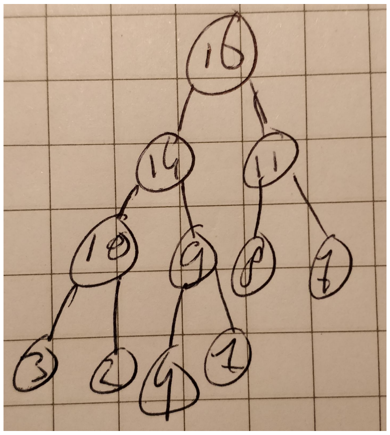
4.1

To sort the array in a way such that we get the Min-heap we first take the root (first element in the array) and put that at the end of our new array. So that the highest value is at the end.
Then for all children of the child of the first node we check which is the highest and we append that at the front of the array.
We do this for all the children of the original array.
This algorithm takes $(O)(?)$.

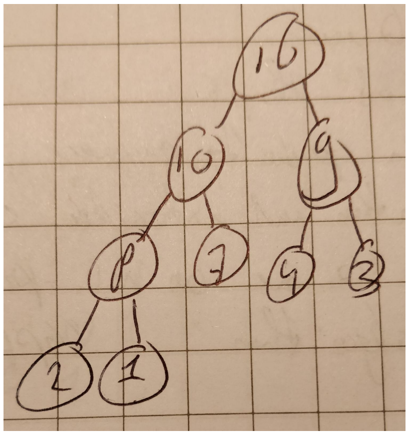
Alternatively we can sort the array from the lowest value to the highest value. This however takes $(O)(n?)$.

4.2

1)



2)

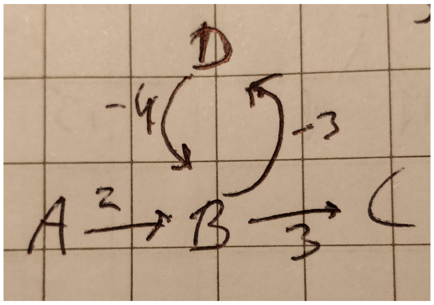


4.3

Q:	0	1	2	3	4	5
	0	30	40	∞	∞	∞
			40	50	80	∞
				50	70	∞
					70	110

4.4

The following graph will not find the shortest path since the theoretical shortest path is $-\infty$.
This is caused by the loop from B -> D that has a negative weight. In this case it will try to find the shortest path by looping through this loop infinitely many times.



4.5

In this case we can create a Graph with the airports as vertices and the distance between them as weights for the edges.
When we have obtained this graph we can run dijkstra's algorithm to get the shortest path from our source vertex s to all other vertices in the graph.
In this case this also gives the values for *easiest*.

4.6

To solve this problem we first run Dijkstra with the chosen v as the source node. This will give us the shortest path from v to all other vertices.
Then we run Dijkstra for all the other vertices and note the shortest path from every vertex to v. ?(Kan

misschien sneller?)
After this we can add those paths to get the shortest path from any chosen vertex to any other vertex via v.
The complexity of this algorithm is $|V| \cdot |V|^2 = |V|^3$ when using arrays.
This complexity can be improved to $|V| \cdot |E| + |V| lg |V|$ when using a Fibonacci heap.

4.7

To solve this problem we need to find the edge e between the cities s and t such that e has the biggest impact on the length of the shortest path from s to t.
We can run Dijkstra's algorithm on the graph with the source node set to s and find the shortest path to t. Then we find the longest edge along this path. This edge is the road that needs to be improved to shorten the path from s to t the most.