Coding Comparison

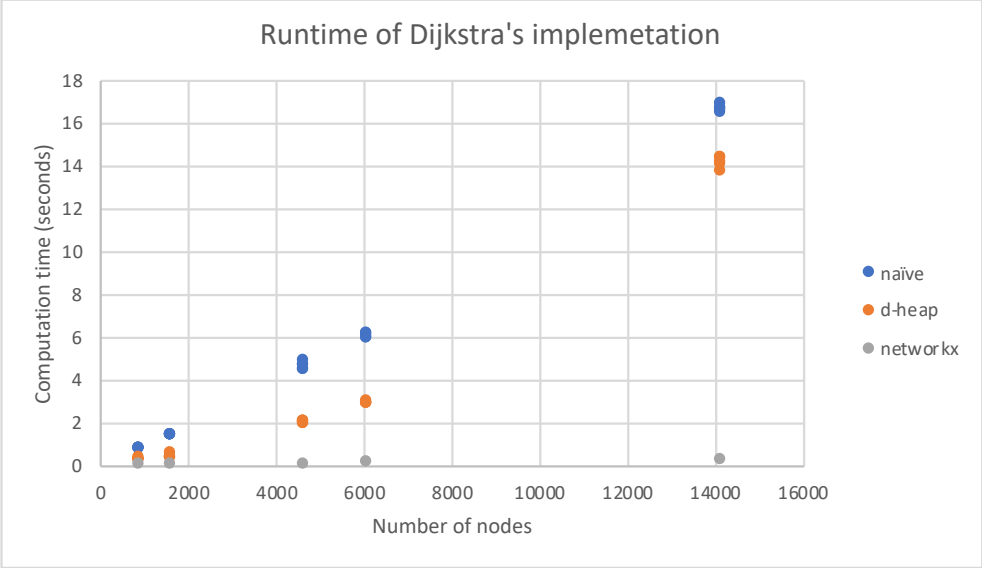
Networks Utilized

The networks I utilized were from the University of Waterloo, Canada, TSP website- http://www.math.uwaterloo.ca/tsp/index.html . The networks are nations – each city (node) in the network has an associated pair of (x,y) position coordinates. The networks do not initially have arcs built in, but a distance matrix between all of the node. I utilized this information to generate a subset of the potential paths between any two nodes. I set the outdegree of every node equal to $\lfloor log\ (n)\ \times 1.2 \rfloor$ - this was taken by trial and error – values smaller than 1.2 generated networks that were not strongly connected. Arcs were generated randomly, by generating random integers in the range of the number of nodes, e.g. other nodes. Arc weights were assigned by taking the Euclidean distance based off of the position coordinates of the head and tail node. The networks are as follows – Canada, 4663 nodes; Morocco, 14185 nodes; Rwanda, 1621 nodes; Tanzania, 6117 nodes; Zimbabwe, 929 nodes.

Discussion

The results show, as expected, that the binary heap implementation of Dijkstra's is a speed up from the naïve Dijkstras – in the 4 smallest cases, it was slightly more than twice as fast. In the largest case, Morocco, binary heap was faster, but only by ~15%. Since binary heap runs in $O(m\ log\ n)$, as the number of arcs dominate the number of nodes; however, one additional reason for slowdown may be that the network had a structure such that more sift down operations occurred (worse in complexity than sift up). Additionally, this code is not fully optimized, as I utilized the pseudocode from Ahuja et. al. to guide the structure of my code. As such there, are performance gains to be made – as indicated by the NetworkX data points. This is the Dijkstra algorithm provided by an the NetworkX package in python, and it ran in 1/100th the time of the naïve Dijkstra's implementation.

Runtime



Runtime Data

Naïve Dijkstra

SOURCE	NODES	ARCS	RUNTIME(S)
3249	4663	46563	4.9
712	4663	46563	4.62
726	4663	46563	4.48
3207	4663	46563	4.43
519	4663	46563	4.59
12305	14185	155979	16.66
6125	14185	155979	16.47
2637	14185	155979	16.85
3057	14185	155979	16.6
7211	14185	155979	16.53
1095	1621	12926	1.4
797	1621	12926	1.41
449	1621	12926	1.4
170	1621	12926	1.37
968	1621	12926	1.38
5593	6117	61113	5.97
4131	6117	61113	6.15
4119	6117	61113	5.92
4166	6117	61113	6.09
7	6117	61113	5.93
859	929	7397	0.77
27	929	7397	0.783
602	929	7397	0.767
730	929	7397	0.766
151	929	7397	0.766

Dijkstra – NetworkX package, python

SOURCE	NODES	ARCS	RUNTIME
3249	4664	46563	0.049
12305	14186	155979	0.18
1095	1622	12926	0.0127
5593	6118	61113	0.064
859	930	7397	0.007

Binary Heap Dijkstra

SOURCE	NODES	ARCS	RUNTIME(S)
3249	4662	46563	2.01
712	4662	46563	1.95
726	4662	46563	1.98
3207	4662	46563	2.03
519	4662	46563	1.93
12305	14185	155979	14.37
6125	14185	155979	14.29
2637	14185	155979	13.98
3057	14185	155979	14.1
7211	14185	155979	13.65
1095	1621	12926	0.51
797	1621	12926	0.37
449	1621	12926	0.36
170	1621	12926	0.36
968	1621	12926	0.36
5593	6117	61113	2.86
4131	6117	61113	2.88
4119	6117	61113	2.8
4166	6117	61113	2.89
7	6117	61113	2.91
859	929	7397	0.178
27	929	7397	0.179
602	929	7397	0.178
730	929	7397	0.18
151	929	7397	0.322

Distance Matrices

Canada - 4663 nodes

	3249	**712**	**726**	**3207**	**519**
3249	0	49327	36487	47417	39189
712	43603	0	28820	51183	22426
726	53861	42505	0	43477	32724
3207	52908	53956	52287	0	45753
519	40491	24744	15711	42330	0

Morocco - 14186 nodes

	12305	6125	2637	3057	7211
12305	0	8718	8936	7644	8060
6125	8092	0	8635	8690	7228
2637	6869		0	8161	4920
3057	8271	5388	7766	0	6916
7211	8618	5655	4001	8676	0

Rwanda -1622 nodes

	1095	797	449	170	968
1095	0	1409	1049	777	1327
797	2047	0	1666	1570	1886
449	1366	1396	0	1542	1730
170	1866	1950	941	0	1838
968	1741	1746	1606	2046	0

Tanzania – 6117 nodes

	5593	4131	4119	4166	7
5593	0	14829	11721	12157	13471
4131	15538	0	14665	15987	11367
4119	14743	18854	0	16399	16871
4166	15076	12336	14459	0	13909
7	15794	16921	13412	14725	0

Zimbabwe – 929 nodes

	859	27	602	730	151
859	0	4268	4251	4184	3667
27	3925	0	3132	3348	3199
602	4080	2323	0	2627	4496
730	4723	2941	3704	0	5112
151	4670	4781	4116	4987	0