

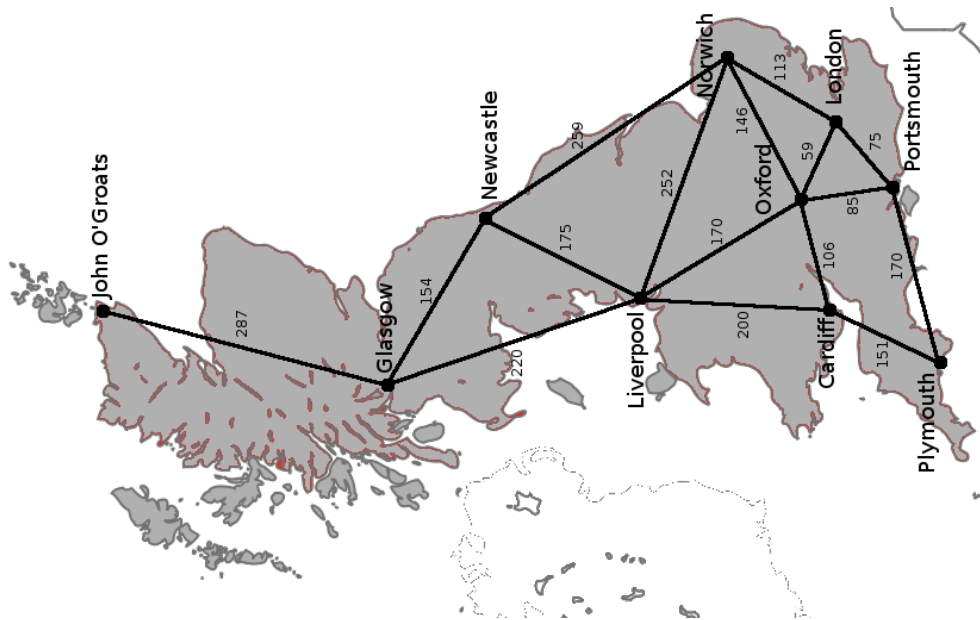
# MS6a, Exercises Week 2

Rune Lyngsø

October 20, 2009

## A Shortest Paths

- Using the following network connecting ten towns in Great Britain, find the shortest path distance from Liverpool to the other nine towns.



- Assume that we are interested in just the shortest path from Liverpool to John O'Groats, and we use the following table of straight line distances to John O'Groats to bias the search by modifying edge weights as described in the lecture notes.

	Cardiff	Glasgow	Liverpool	London	Newcastle	Norwich	Oxford	Plymouth	Portsmouth
Distance to John O'Groats	491	191	358	506	258	450	479	570	547

What is the modified edge weight between Liverpool and Norwich? Using Dijkstra's algorithm on the road network with modified weights, how many of the other eight towns would be extracted from the priority queue before John O'Groats (i.e. how many towns are closer to Liverpool than John O'Groats with the modified edge weights)? How many were closer to Liverpool using the original distances?

- Give an example of a graph with one or more negative edge weights where Dijkstra's algorithm fail.
- The following table specifies an alternative splicing model for Ensembl gene ENSG0000081052, based on the alternative splicing graph available from <http://statgen.ncsu.edu/asg/>. Each node corresponds to

a segment of DNA, and there is an edge connecting two nodes if they can be spliced together (i.e. the entire intervening segment can be removed when the splicing machinery constructs a mature messenger RNA from the RNA transcribed from the gene). The end of a segment is called a donor site and the beginning of a site an acceptor site. In our model we assume that each donor site has a probability distribution the acceptor site is chosen from; only acceptor sites there is an edge to can be chosen. In short, from each node we choose the next node according to a probability distribution on the outgoing edges. A splice form is a full path from the first node (which does not correspond to an actual segment, but is used to model the possibility of alternative start positions) to the last node (again not an actual segment, but there to allow alternative end positions – this also explains why this graph has 44 nodes while the one in the alternative splicing gallery only has 42).

0	1 : 0.5	3 : 0.25	24 : 0.25	22	23 : 1
1	2 : 0.5	4 : 0.5		23	25 : 0.25 26 : 0.5 32 : 0.25
2	5 : 1			24	25 : 1
3	4 : 1			25	26 : 0.5 28 : 0.5
4	5 : 1			26	27 : 0.34 31 : 0.33 38 : 0.33
5	6 : 0.25 7 : 0.25 8 : 0.25 11 : 0.25			27	28 : 1
6	7 : 1			28	29 : 1
7	8 : 1			29	30 : 0.66 31 : 0.34
8	9 : 0.75 11 : 0.25			30	31 : 1
9	10 : 0.5 11 : 0.5			31	32 : 1
10	11 : 1			32	33 : 0.34 34 : 0.33 35 : 0.33
11	12 : 0.66 13 : 0.17 16 : 0.17			33	35 : 1
12	13 : 0.5 14 : 0.25 15 : 0.25			34	35 : 1
13	14 : 1			35	36 : 0.34 37 : 0.33 39 : 0.33
14	15 : 1			36	37 : 1
15	16 : 1			37	38 : 1
16	17 : 0.43 18 : 0.43 20 : 0.14			38	39 : 1
17	18 : 1			39	40 : 1
18	19 : 0.32 20 : 0.17 21 : 0.17 22 : 0.17 39 : 0.17			40	41 : 0.28 42 : 0.72
19	20 : 1			41	42 : 1
20	21 : 1			42	43 : 1
21	22 : 0.6 23 : 0.2 29 : 0.2			43	

In this table, for each node  $u$  there is a list of  $v : p$  values, where  $v$  is a node there is an edge to and  $p$  the probability of choosing that edge. What is the most probable splice form for this alternative splicing model, i.e. the most likely path from node 0 to node 43? How is this related to shortest paths?

## B Spanning Trees

- Find a minimum spanning tree for the (unmodified) road network in question 1.
- In both the Dijkstra and the Bellman-Ford algorithm we construct a shortest path to node  $u$  by extending a shortest path to a predecessor node  $v$  with edge  $(v, u)$ . So the path  $s = u_0 \rightarrow u_1 \rightarrow \dots \rightarrow u_{k-1} \rightarrow u_k = u$  is the shortest path identified for node  $u$  iff  $s = u_0 \rightarrow u_1 \rightarrow \dots \rightarrow u_{k-1}$  is the shortest path identified for node  $u_{k-1}$ . Remembering the predecessor  $p[u]$  for each node will thus allow us to not only report the distance to a node  $u$ , but also a path of that length – starting from  $u$ , we just keep following the predecessors until we reach  $s$ . How would the Dijkstra algorithm in the lecture notes have to be modified to also tabulate a predecessor for each node  $u$ ?
- Each node only has one predecessor, as defined in question 6, though one node can be the predecessor of several other nodes. The edges  $(p[u], u)$  thus constitutes a spanning tree rooted at  $s$  for a graph, and for each node the length of the path to  $s$  in this tree is minimal. Is the tree a minimum spanning tree? If yes, argue why, if no, give a counter example of a graph  $G$  and a starting node  $s$  such that the tree of shortest paths  $s$  is not a minimum spanning tree.