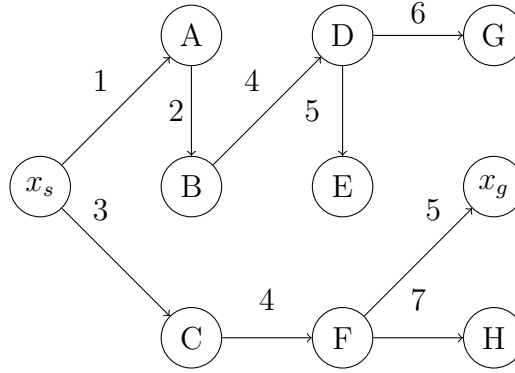


1 Dijkstra's algorithm

Consider the following graph. Let the start vertex be x_s and the goal vertex be x_g . Important: if a node has multiple outgoing edges, when the node is expanded the vertices adjacent to the node are processed in alphabetical order. Similarly, if multiple nodes in the OPEN queue have the same priority value, sort them by alphabetical order.

Step	x_s	A	B	C	D	E	F	G	H	x_g
0	N/0	∞	∞	∞	∞	∞	∞	∞	∞	∞
1	N/0	$x_s/1$	$x_s/5$	$x_s/3$	∞	∞	∞	∞	∞	∞
2	N/0	$x_s/1$	A/2	$x_s/3$	A/5	B/9	C/4	∞	∞	∞
3	N/0	$x_s/1$	A/2	$x_s/3$	B/4	B/6	C/4	∞	∞	∞
4	N/0	$x_s/1$	A/2	$x_s/3$	B/4	D/5	C/4	D/6	F/7	∞
5	N/0	$x_s/1$	A/2	$x_s/3$	B/4	D/5	C/4	D/6	F/7	F/5



2 A*

Consider the directed weighted graph shown at page 2. Show how the algorithm A* would determine the shortest path between x_s and x_g . Follow the same method used in class and in example 4.5 in the lecture notes.

OPEN	x_s	A	B	C	D	E	F	G	x_g
x_s	N/0/7	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞
F,A,B	x_s	$x_s/2/14$	$x_s/3/14$	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	$x_s/5/11$	N/ ∞ / ∞	N/ ∞ / ∞
F,A,B	x_s	$x_s/2/14$	$x_s/3/14$	N/ ∞ / ∞	N/ ∞ / ∞	N/ ∞ / ∞	$x_s/5/11$	N/ ∞ / ∞	N/ ∞ / ∞
F	x_s	$x_s/2/14$	$x_s/3/14$	F/7/14	N/ ∞ / ∞	F/9/13	$x_s/5/11$	N/ ∞ / ∞	N/ ∞ / ∞
E, C	x_s	$x_s/2/14$	$x_s/3/14$	F/7/14	E/10/14	F/9/13	$x_s/5/11$	E/16/14	C/17/14
D	x_s	$x_s/2/14$	$x_s/3/14$	F/7/14	E/10/14	F/9/13	$x_s/5/11$	E/16/14	D/12/14

3 Navigation functions 1

8	7	6	5	4	3	2	1	2	3
7	6	5	4	3	2	1	0	1	2
8	7	6	5	4	3	2	1	2	3
		7	6	5			2	3	4
10	9	8	7	6	5	4	3	4	5
11	10	9	8			5	4	5	6
12						6	5	6	7
13						7	6	7	8
14	13	12	11	10	9	8			9
15	14	13	12	11	10	9	10	11	10

4 Navigation functions 2

1. Visually, the greatest offender of the navigation function is that bottom left the numbers jump from 10 to 15. There are other instances where the numbers are not adding up properly like on the left side again with the 10 and 7. There are also the bottom right as 6 goes to 9 then 8 which is wrong.

2.

6	5	4	3	2	1	2	3
7	6		2	1	0		
8	7		3	2	1	2	3
9	8			3	2	3	3
10	9		5	4	3	4	5
9	10		6	5	4	5	6
10	9	8	7	6	5		7
11	10	9	8	7	6	7	8