University of Nevada, Las Vegas Computer Science 477/677 Spring 2024

Assignment 5: Due Saturday March 30 2024 11:59 PM

Upload your homework to Canvas.

Name:_____

You are permitted to work in groups, get help from others, read books, and use the internet. Your answers must be written in a pdf file and uploaded to canvas, by midnight February 16th. Your file must not be unnecessarily long. If you have any questions, or you are having trouble uploading the assignment you may email the grader, Sebrina Wallace, at `wallace4@unlv.nevada.edu`. You may also send me email to ask questions.

For each shortest path problem, let n be the number of vertices, and m the number of arcs.

1. What is the worst case asymptotic time complexity for each of these shortest path algorithms?
   (a) Bellman Ford _____
   (b) Floyd Warshall _____
   (c) Dijkstra _____
   (d) Johnson _____
2. Write pseudocode for the Floyd Warshall algorithm. Let W[i,j] be the length of the arc from i to j, which could be ∞.

3. Write pseudocode for the Bellmain Ford algorithm. Be sure to include the shortcut.

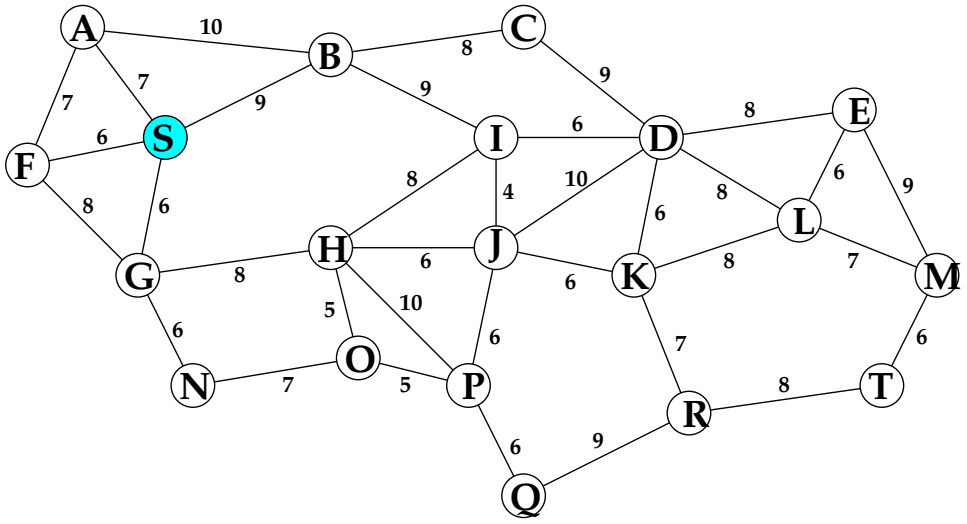
4. Walk through Dijkstra's algorithm for the single source minpath problem for the directed graph illustrated below. Instead of numbering the vertices 0 through 19, I have assigned them letters from A to T. The source vertex is S.

   After each iteration of the main loop, show

   1. The array dist, where dist[x] is the smallest length of any path found so far from S to x. (Initially, dist[x] = ∞ for most x.)

   2. The array back, where back[x] is the next-to-the last vertex on the path of smallest weight found so far from S to x.

   3 The contents of heap. Do not try to show the structure of the heap, simply list its members.



5. Write C++ code (which must be able to be executed) for a dynamic program which finds the maximum weight legal subsequence of a sequence of positive integers. A subsequence is legal if it has no consecutive terms of the input sequence. Your program should use backpointers to recover the best subsequence.

   In order to make the grader's task easier, here are two sequences I want you to use while testing your program, each of length 20.

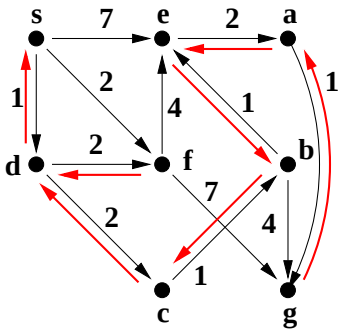
   15,324,184,48,102,31,119,15,26,160,129,78,14,7,64,83,22,88,185,123.

   139,79,19,53,52,23,94,298,209,59,16,116,80,130,279,195,6,37,161,206.

   Remember that the task of this assignment is not just to find the correct answers, but to write and run a correct program.

Dijkstra's Algorithm Practice Problems and Solutions

1. Use Dijkstra's algorithm to solve the single source shortest path problem for the following weighted directed graph, where **s** is the source. Show the steps.

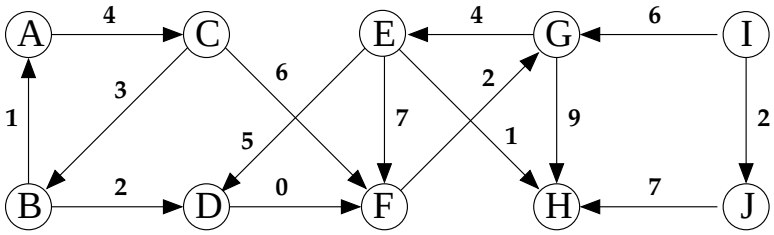


s	a	b	c	d	e	f	g
0	7	4	3	1	~~7~~ 5	2	~~9~~ 8
*	e	c	d	s	~~s~~ b	s	~~f~~ b

Let Q be the minqueue consisting of partially processed vertices. We write Q as a list of vertices, sorted by V. Initially, $Q = (\mathbf{s})$.

In the first step, we dequeue **s** and enqueue **d,e,f**. $Q = (\text{d,f,e})$
In the second step we dequeue **d** and enqueue **c**. $Q = (\text{f,c,e})$
In the third step, we dequeue **f**, enqueue **g**, and change **back(e)** to **f**, decreasing **V(e)** from 7 to 6. $Q = (\mathbf{c,e,g})$
In the fourth step, we dequeue **c** and enqueue **b**. Now, $Q = (\mathbf{b,e,g})$.
In the fifth step, we dequeue **b**. $Q = (\mathbf{e,g})$, and change **back(g)** to **b**, decreasing **V(g)** from 9 to 8. $Q = (\mathbf{e,g})$.
In the sixth step, we dequeue **e** and enqueue **a**. $Q = (\mathbf{a,g})$.
In the seventh step, we dequeue **a**. $Q = (\mathbf{g})$.
In the eighth and last step, we dequeue **g**. Q is now empty and we are done.

2. Let G be the directed graph given below. Use Dijkstra's algorithm to solve the single source shortest path problem on G with start vertex A. Show your work.



I will try to use the notation used in the code on page 110 of our textbook, Thus, the backpointer is called prev. The second array is the minqueue, Q, where minimum dist node is on the left. Initially, A is the only member of the queue, and its backpointer prev[A] is undefined.

	A	B	C	D	E	F	G	H	I	J
dist	0	∞	∞	∞	∞	∞	∞	∞	∞	∞
prev	$\star$									

A
0

At each step, we execute u = deletemin(Q). Thus u = A. We insert C into Q, since it is the only outneighbor of A.

	A	B	C	D	E	F	G	H	I	J
dist	0	∞	4	∞	∞	∞	∞	∞	∞	∞
prev	$\star$		A							

C
4

Now we let u = deletemin(Q) = C, and we insert B and F into Q.

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	∞	∞	10	∞	∞	∞	∞
prev	$\star$	C	A			C				

B	F
7	10

Now we let u = deletemin(Q) = B, and we insert D into Q. Note that D is ahead of F in the priority.

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	9	∞	10	∞	∞	∞	∞
prev	$\star$	C	A	B		C				

D	F
9	10

Now we let u = deletemin(Q) = D, and we update dist(F) and prev(F).

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	9	∞	9	∞	∞	∞	∞
prev	$\star$	C	A	B		D				

F
9

Now we let u = deletemin(Q) = F, and we insert G into Q. Now we let u = deletemin(Q) = F, and we insert G into Q.

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	9	∞	9	11	∞	∞	∞
prev	$\star$	C	A	B		D	F			

G
11

Now we let u = deletemin(Q) = G, and we insert E and H into Q.

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	9	15	9	11	20	∞	∞
prev	$\star$	C	A	B	G	D	F	G		

E	H
15	20

Now we let u = deletemin(E) = G, and we update dist[H] and prev[H]

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	9	15	9	11	16	∞	∞
prev	$\star$	C	A	B	G	D	F	E		

H
16

Now we let u = deletemin(E) = H. Since H has no outneighbors, we do not insert anything into Q. Since Q is empty, we are done. I and J are unreachable from A.

	A	B	C	D	E	F	G	H	I	J
dist	0	7	4	9	15	9	11	16	∞	∞
prev	$\star$	C	A	B	G	D	F	E		