University of Nevada, Las Vegas Computer Science 477/677 Fall 2025

Answers to Assignment 4: Due Saturday October 18, 2025

Follow our TA Louis DuMontet's (`dumontet@unlv.nevada.edu`) instructions on how to turn in the assignment.

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet.

1. True/False/Open

   (a) **F** Every mathematically defined problem can be solved by computation, although it might not have been done yet.

   (b) **O** "Many hands make light work." Any polynomial time problem can be worked in $O(\log^k n)$ time, for some k, by polynomially many processers working in parallel.

   (c) **F** Computers are so fast nowadays that analyzing algorithms for time complexity is pointless.

2. Write pseudocode for the Bellman-Ford algorithm. Be sure to include the shortcut. Assume that the vertices are numbered from 0 to $n-1$ with source vertex 0, and that the arcs are numbered from 0 to $m-1$, and that the j^{th} arc is $a_j = (s_j, t_j)$ for vertices s_j and t_j, and has weight w_j. Be sure to compute backpointers.

   Let $V[i]$ be the least cost path from 0 to i found so far.

   for $i = 1$ to $n-1$
       $V[i] = \infty$
       $V[0] = 0$
       finished = false
       while not finished
           finished = true
           for $j = 0$ to $m-1$
               temp $= V[s_j] + w_j$
               if temp $< V[t_j]$
                   $V[t_j] =$ temp
                   finished = false
                   back$(t_j) = s_j$

3. Solve the following recurrences. Express the answers using Θ notation.

(a) $F(n) = 3F(2n/3) + 3F(n/3) + n^3$

$F(n) = \Theta(n^3 \log n)$

(b) $F(n) = F(n/2) + 2F(n/4) + 1$

$F(n) = \Theta(n)$

(c) $F(n) = F(n/2) + F(n/3) + n$

$F(n) = \Theta(n)$

(d) $F(n) = 9F(n/3) + n$

$F(n) = \Theta(n^2)$

(e) $F(n) = 3F(n/3) + n;$

$F(n) = \Theta(n \log n)$

(f) $F(n) = F(n/3) + 1;$

$F(n) = \Theta(\log n)$

(g) $F(n) = F(n-1) + \log n;$

$F(n) = \Theta(n \log n)$

(h) $F(n) = F(n - \sqrt{n}) + \sqrt{n};$

$F(n) = \Theta(n)$

(i) $F(n) = F(n - \sqrt{n}) + 1$

$F(n) = \Theta(\sqrt{n})$

(j) $F(n) = F(3n/5) + F(4n/5) + n^2$

$F(n) = \Theta(n^2 \log n)$

(k) Duplicate eliminiated

(l) $F(n) = F(12n/13) + F(5n/13) + n$

$F(n) = \Theta(n^2)$

(m) $F(n) = 2F(n-1) + 1$

$F(n) = \Theta(2^n)$

4. The following code computes the product of its two parameters. Write the loop invariant of the loop.

```
float product(float x, int n)
  // input condition: n >= 0
 {
  float y = x;
  int m = n;
  float z = 0.0;
  while(m > 0)
   {
    if(m%2) z = z+y;
    y = y+y;
    m = m/2;
   }
  return z;
 }
```

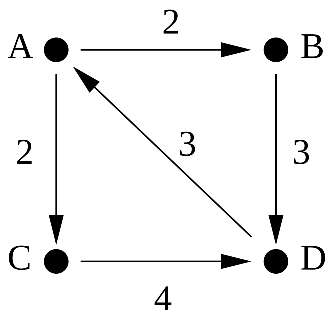
$n * x = m * y + z$

5. Write pseudocode for the Floyd-Warshall algorithm. Assume that the vertices are numbered from 1 to n, that W(i,j) is the weight of the arc from i to j, and that W(i,j) $= \infty$ if there is no arc from i to j. Be sure to compute back pointers.

6. Use heapsort to sort the array QWERTYUIOP. Use the following matrix for your computation. Add extra rows as necessary.

Q	W	E	R	T	Y	U	I	O	P
Q	W	Y	R	T	E	U	I	O	P
Y	W	Q	R	T	E	U	I	O	P
Y	W	U	R	T	E	Q	I	O	P
P	W	U	R	T	E	Q	I	O	**Y**
W	P	U	R	T	E	Q	I	O	**Y**
W	T	U	R	P	E	Q	I	O	**Y**
O	T	U	R	P	E	Q	I	**W**	**Y**
U	T	Q	R	P	E	O	I	**W**	**Y**
I	T	Q	R	P	E	O	**U**	**W**	**Y**
T	I	Q	R	P	E	O	**U**	**W**	**Y**
T	R	Q	I	P	E	O	**U**	**W**	**Y**
O	R	Q	I	P	E	**T**	**U**	**W**	**Y**
R	O	Q	I	P	E	**T**	**U**	**W**	**Y**
R	P	Q	I	O	E	**T**	**U**	**W**	**Y**
E	P	Q	I	O	**R**	**T**	**U**	**W**	**Y**
Q	P	E	I	O	**R**	**T**	**U**	**W**	**Y**
O	P	E	I	**Q**	**R**	**T**	**U**	**W**	**Y**
P	O	E	I	**Q**	**R**	**T**	**U**	**W**	**Y**
I	O	E	**P**	**Q**	**R**	**T**	**U**	**W**	**Y**
O	I	E	**P**	**Q**	**R**	**T**	**U**	**W**	**Y**
E	I	**O**	**P**	**Q**	**R**	**T**	**U**	**W**	**Y**
I	E	**O**	**P**	**Q**	**R**	**T**	**U**	**W**	**Y**
E	**I**	**O**	**P**	**Q**	**R**	**T**	**U**	**W**	**Y**
E	**I**	**O**	**P**	**Q**	**R**	**T**	**U**	**W**	**Y**

7. (a) Finish the matrix W representing the weighted directed graph shown below. I have written the first two rows. (b) Use tropical matrix multiplication to compute W^3, which represents the graph of minimum path distances. Do not calculate backpointers.



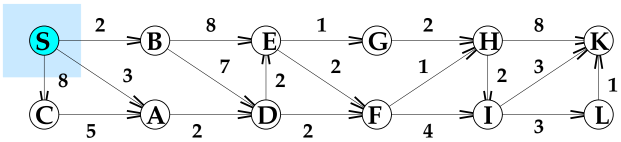
$$W = \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & \infty \\ B & \infty & 0 & \infty & 3 \\ C & \infty & \infty & 0 & 4 \\ D & 3 & \infty & \infty & 0 \\ \hline \end{array}$$

$$W^2 = \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & \infty \\ B & \infty & 0 & \infty & 3 \\ C & \infty & \infty & 0 & 4 \\ D & 3 & \infty & \infty & 0 \\ \hline \end{array} \; \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & \infty \\ B & \infty & 0 & \infty & 3 \\ C & \infty & \infty & 0 & 4 \\ D & 3 & \infty & \infty & 0 \\ \hline \end{array} = \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & 5 \\ B & 6 & 0 & \infty & 3 \\ C & 7 & \infty & 0 & 4 \\ D & 3 & 5 & 5 & 0 \\ \hline \end{array}$$

$$W^3 = \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & \infty \\ B & \infty & 0 & \infty & 3 \\ C & \infty & \infty & 0 & 4 \\ D & 3 & \infty & \infty & 0 \\ \hline \end{array} \; \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & 5 \\ B & 6 & 0 & \infty & 3 \\ C & 7 & \infty & 0 & 4 \\ D & 3 & 5 & 5 & 0 \\ \hline \end{array} = \begin{array}{c|cccc|} & A & B & C & D \\ \hline A & 0 & 2 & 2 & 5 \\ B & 6 & 0 & 8 & 3 \\ C & 7 & 9 & 0 & 4 \\ D & 3 & 5 & 5 & 0 \\ \hline \end{array}$$

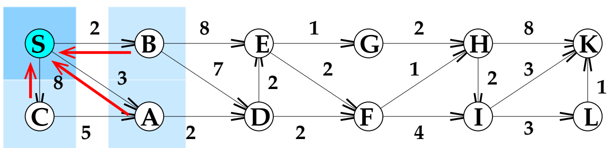
Think! What would W^4 represent?

8. Solve the single source minpath problem using Dijkstra's algorithm for the weighted graph shown, where S is the source vertex. Fill in the matrix at each step, as was done for the example shown in the handout shown in class. Each edge shown is actually two arcs, one in each direction. Attach extra sheets if needed.



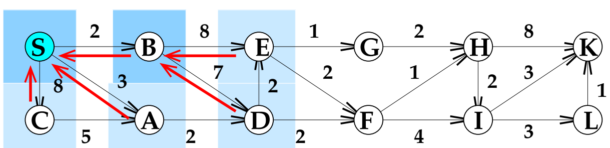
	S	A	B	C	D	E	F	G	H	I	K	L
V	0											
back												

Minheap = S



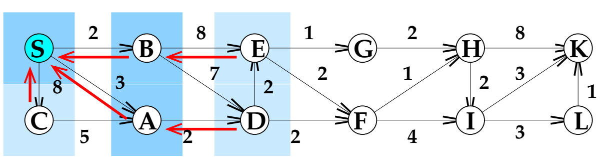
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8								
back		S	S	S								

Minheap = BAC



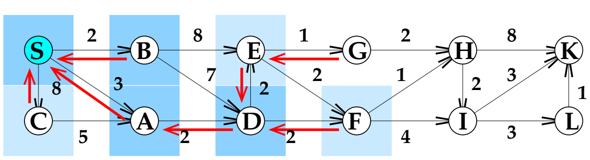
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	9	10						
back		S	S	S	B	B						

Minheap = ACDE



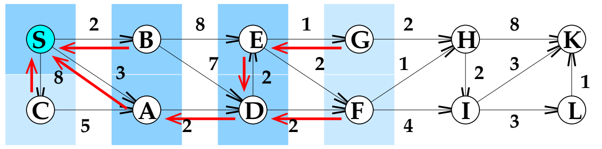
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	10						
back		S	S	S	~~B~~ A	B						

Minheap = DCE



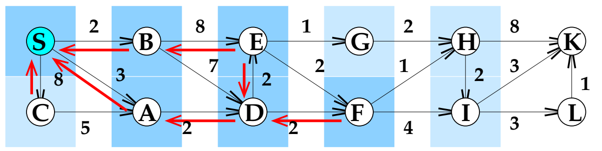
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	~~10~~ 7	7					
back		S	S	S	~~B~~ A	~~B~~ D	D					

Minheap = FCE



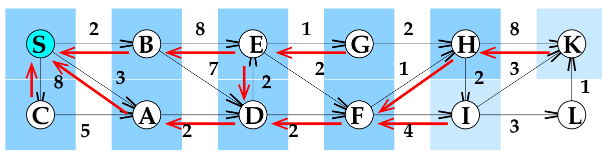
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	~~10~~ 7	7	8				
back		S	S	S	~~B~~ A	~~B~~ D	D	E				

Minheap = FCG



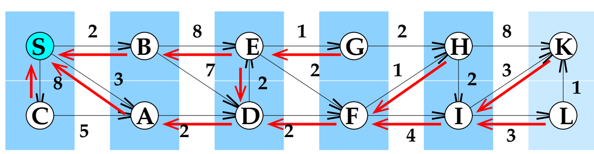
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	~~10~~ 7	7	8	8	11		
back		S	S	S	~~B~~ A	~~B~~ D	D	E	F	F		

Minheap = CGHI



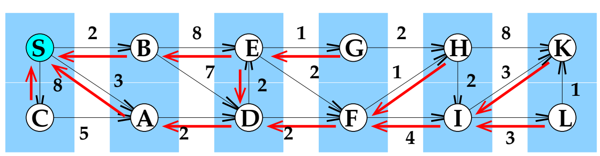
	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	~~10~~ 7	7	8	8	11	16	
back		S	S	S	~~B~~ A	~~B~~ D	D	E	F	F	H	

Minheap = IK



	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	~~10~~ 7	7	8	8	11	~~16~~ 14	14
back		S	S	S	~~B~~ A	~~B~~ D	D	E	F	F	~~H~~ I	I

Minheap = KL



	S	A	B	C	D	E	F	G	H	I	K	L
V	0	3	2	8	~~9~~ 5	~~10~~ 7	7	8	8	11	~~16~~ 14	14
back		S	S	S	~~B~~ A	~~B~~ D	D	E	F	F	~~H~~ I	I

Minheap = ∅