University of Nevada, Las Vegas Las Vegas Computer Science 477/677 Spring 2020

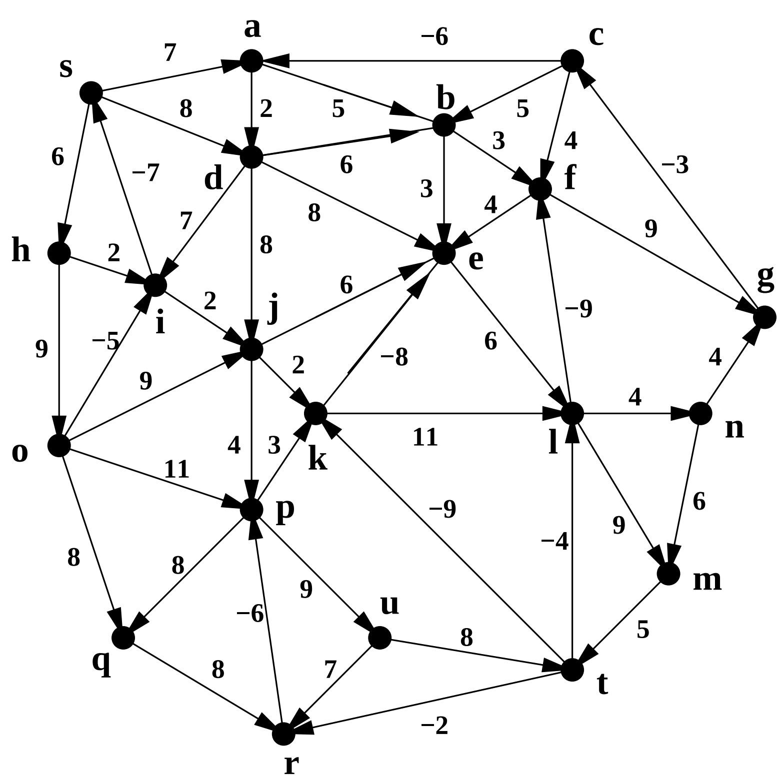
Assignment 6: Due Thursday April 9, 2020

Name:____

You are permitted to work in groups, get help from others, read books, and use the internet. But the handwriting on this document must be your own. Print out the document, staple, and fill in the answers. You may attach extra sheets, but only by stapling. ~~Turn in the pages to the graduate assistant at the beginning of class, April 7.~~

We will not meet in the classrom on April 9. There are instructions for turning in your homework on the assignments page.

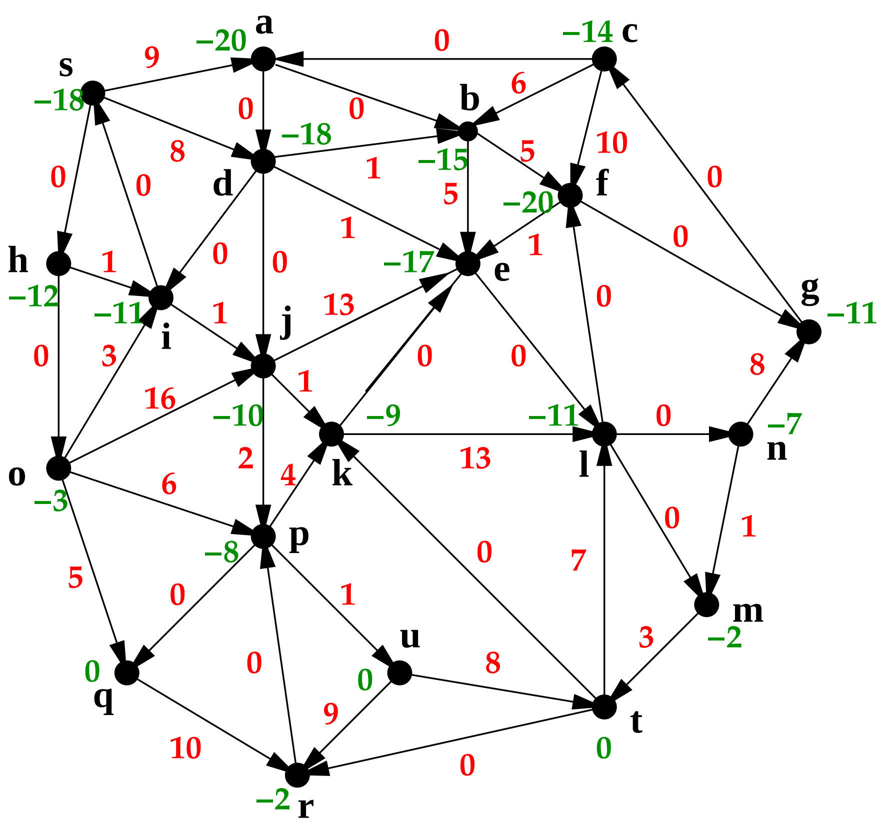
1. Consider the all pairs shortest path problem on the weighted directed graph shown below in (a). The first step of Johnson's algorithm is to compute a "heuristic" $h(x)$ for each vertex x. In Figure (b), indicate the value of h at every vertex. Recall that $h(x) \leq 0$ for all x.



(a)

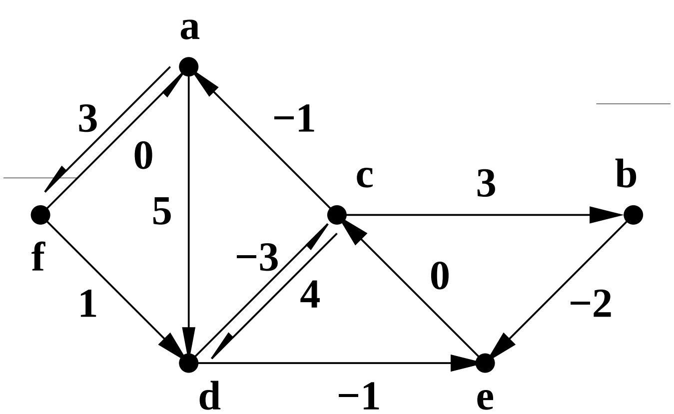
The second step of Johnson's algorithm is to compute an "adjusted" weight on each edge. That adjusted weight is always non-negative. Also in Figure (b), indicate that adjusted weight on each edge.

The heuristic value at each vertex is in green. The adjusted weight at each edge is in red.



(b)

2. Let G be the weighted directed graph shown in the figure below. Assume you execute an algorithm for the all pairs problem on G. The ouput of that program consist of two $n \times n$ matrices, where $n = 6$. Fill in the matrices for that output. (You are not required to use any particular algorithm; you can even try to "eye-ball" the answers, if you **dare**.)

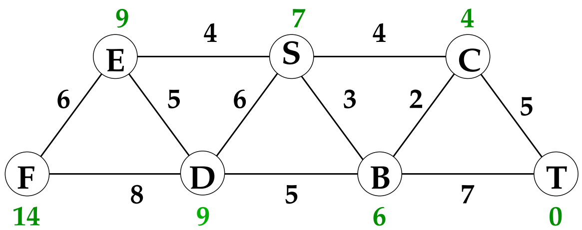


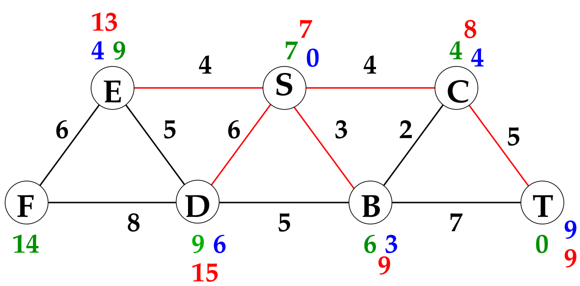
	a	b	c	d	e	f
a	0	4	1	4	3	3
b	−3	0	−2	1	−2	0
c	−1	3	0	3	2	2
d	−4	0	−3	0	−2	−1
e	−1	3	0	3	0	2
f	0	1	−2	1	0	0

	a	b	c	d	e	f
a	*	c	d	f	b	a
b	c	*	e	f	b	a
c	c	c	*	f	b	a
d	c	c	d	*	b	a
e	c	c	e	f	*	a
f	f	c	d	f	b	*

The first array contains the lengths of the shortest path between each pair; the second array contains the backpointers.

3. Find the minimum weight path from S to T for the weighted graph given below, by walking through the $A^\star$ algorithm. Use the technique shown in the Youtube video. The heuristics are given by the green numerals.





In the second graph, we show $g(x)$ in blue, and we show $f(x) = g(x) + h(x)$ in red. Initially, there is only one visited vertex, S, the only member of the minheap, and $g(\mathrm{S}) = 0$, hence $f(\mathrm{S}) = 7$.

S is deleted from the minheap, and its neighbors B, C, D, and E are inserted. The key of the minheap is the value of f, indicated by the red numeral.

In the next step, C is deleted, and its neighbors are examined. There is no decrease in $g(\mathrm{B})$, but $g(\mathrm{T})$ becomes 9.

In the next step, two vertices have equal priority, B and T, both of which have f value 9. If we delete B first, there will be no update and no change in $g(\mathrm{T})$. Thus that move will have no effect.

We will then delete T. The value of $g(\mathrm{T}) = 9$, and we are done.

The edges used for backpointers are indicated in red.

We could construct two arrays, as in the previous problem, where the first array shows minimum path costs using the adjested weights and the second array shows backpointers.

But we really want the results for the original (black) weights. It turns out that the backpointers for the red weights are exactly the same as for the black weights. Thus, using backpointers, we could fill in the first matrix for We could construct two arrays, as in the previous problem, where the first array shows minimum path costs using the adjested weights and the second array shows backpointers.

But, we really want the results for the original (black) weights. The backpointers for the original problem are the same as the backpointers for the adjusted problem. Using those, we could compute the (black) cost of each minimum path.