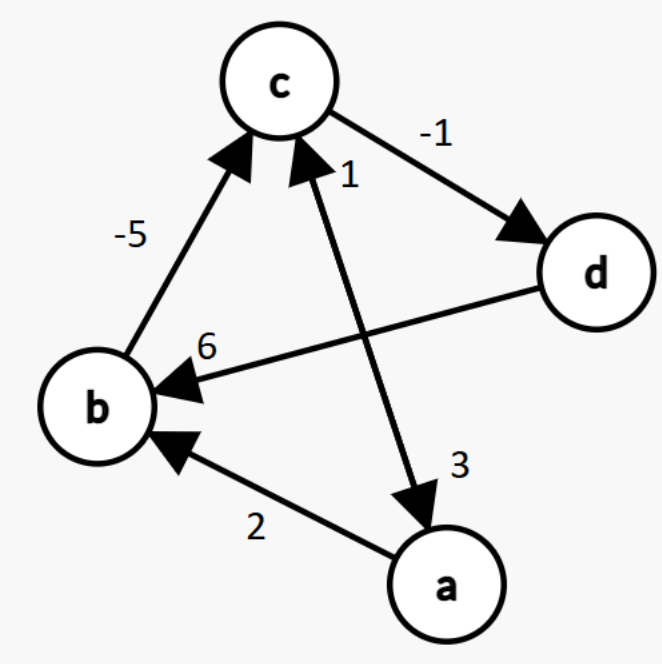
Homework #9

Instructor: Ali Sharifian

Points: 20

For due date, please see Canvas.

Execute the Bellman Ford algorithm on the below network to provide the list of distances and previous nodes from the source. Assume node a is the source node.



Note that there are 4 nodes and 6 edges in the above graph. The link between c and a is bi-directional, so that counts as two edges. Also, to clarify because the above image may not be fully clear:

c->a has an edge weight of 3

a->c has an edge weight of 1

d->b has an edge weight of 6

To understand the type of work and final answer that I'm expecting, see the **Sample Bellman Ford Problem and Solution.pdf** document posted on Canvas.

Node	Distance from a	Previous Node
a	0 ~~infinity~~	Nil
b	2 ~~inifinity~~	a ~~Nil~~
c	-3 ~~inifiniy~~	b ~~Nil~~
d	-4 ~~inifinity~~	c ~~Nil~~

Iteration 1:

a->b:

Is infinity > 0+2? Yes, so update b's distance entry to 0+2=2. Update previous node to a.

b->c:

Is infinity > 2+(-5)? Yes, so update c's distance entry to 2+(-5)=-3. Update previous node to b.

c->d:

Is infinity > (-3) + (-1)? Yes, so update d's distance entry to (-3) + (-1) = -4. Update previous node to c.

d->b:

Is 2>(-4) + 6? No, so don't update row b.

a->c:

Is (-3) > 0 + 1? No, so don't update row c.

c->a:

Is 0 >(-3) + 3? No, so don't' update row a.

Iteration 2:

a-> b: 0+2 = 2, so no update

a-> c: 0+1 > -3, so no update

b -> c: 2+(-5) = -3, so no update

c -> a: -3+ 3 = 0, so no update

c-> d: -3 + (-1) = -4, so no update

d -> b: -4 + 6 = 2, so no update.

Iteration 3:

a->b: 0+2 = 2, so no update

a->c: 0+1 > -3, so no update

b->c: 2+(-5) = -3, so no update

c->a: -3+3 = 0, so no update

c->d: -3 + (-1) = -4, so no update

d-> b: -4+6 = 2, so no update