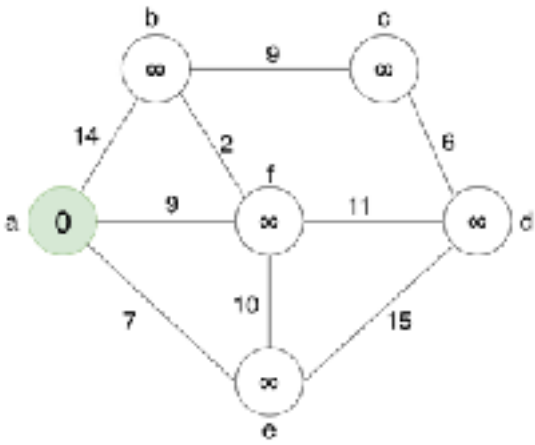
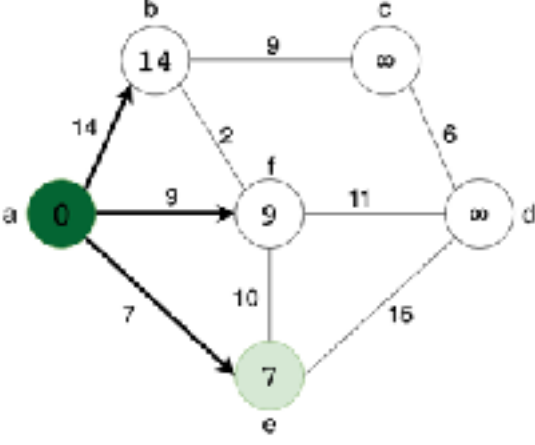
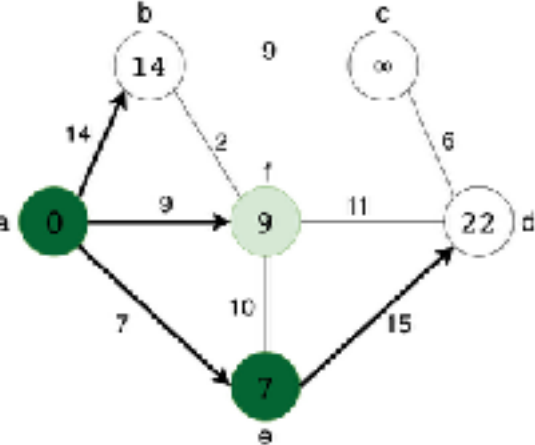
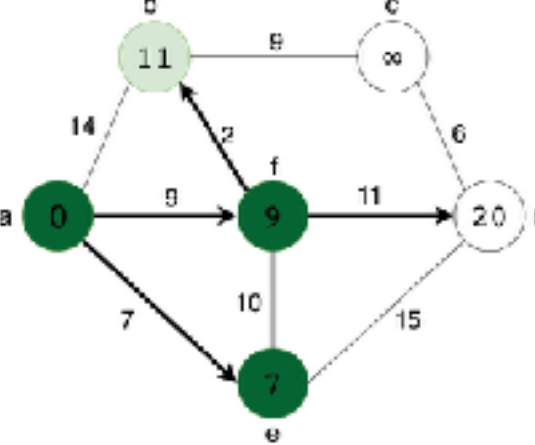
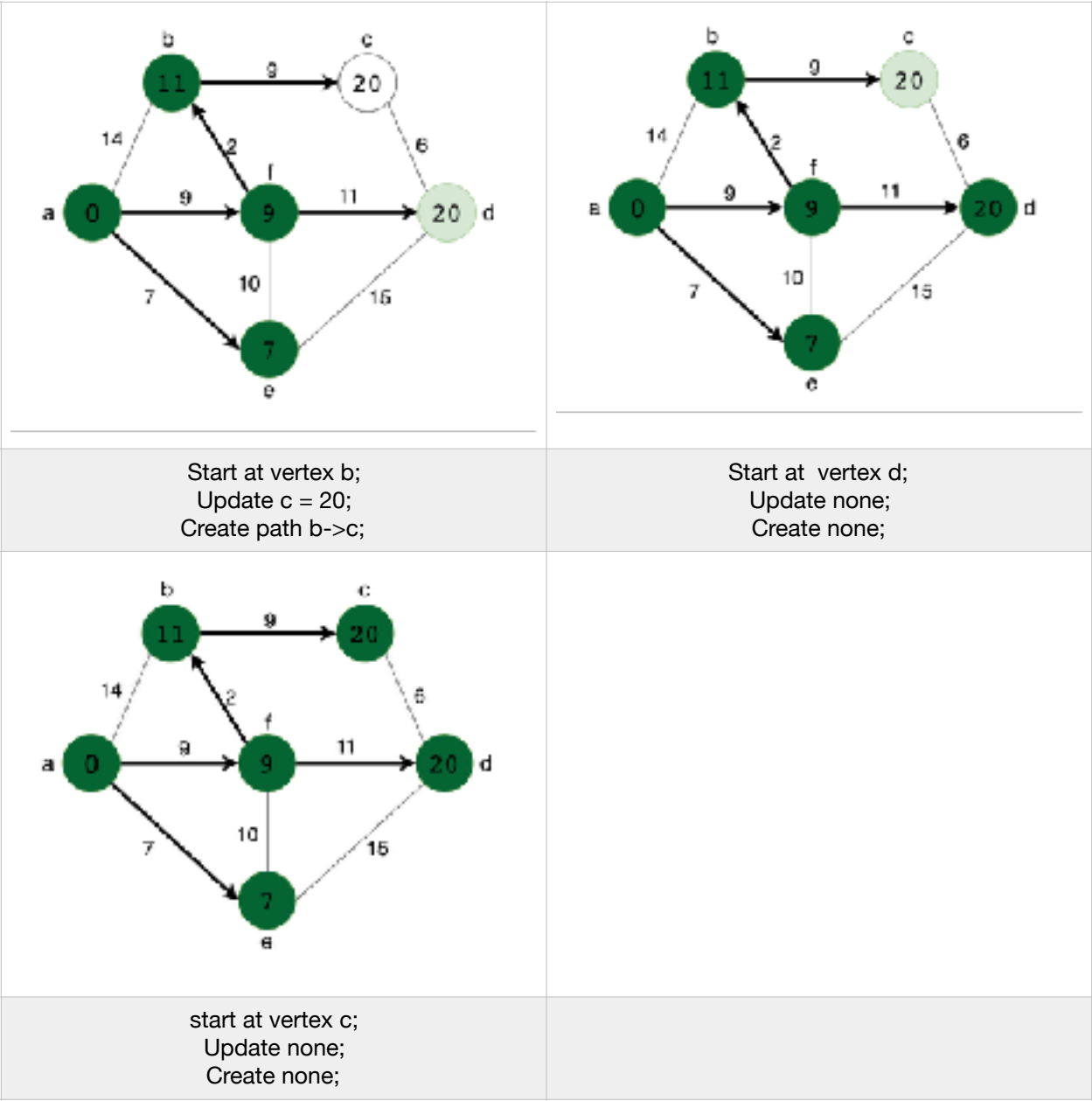


Homework 9

Yaqiao Liu
NUID 001279512

Q1

	
<p>$v[0].distance = 0;$</p>	<p>Start at vertex a; Update distance b,f,e = 14, 9, 7; Create path a->b, a->f, a->e;</p>
	
<p>Start at vertex e; Update distance d = 22; Create path: e->d;</p>	<p>Start at vertex f; Update distance b = 11, d = 20; Create path: f->b, f->d;</p>



Q2

(a) initial state

Distance

	A	B	C	D
A	0	2	6	4

B	2	0	3	∞
C	6	3	0	3
D	4	∞	3	0

Parent

	A	B	C	D
A	NIL	0	0	0
B	1	NIL	1	NIL
C	2	2	NIL	2
D	3	NIL	3	NIL

(b) $k = 0$

Distance

	A	B	C	D
A	0	2	6	4
B	2	0	3	6
C	6	3	0	3
D	4	6	3	0

Parent

	A	B	C	D
A	NIL	0	0	0
B	1	NIL	1	0
C	2	2	NIL	2
D	3	0	3	NIL

(c) $k = 1$

Distance

	A	B	C	D
A	0	2	5	4
B	2	0	3	6
C	5	3	0	3
D	4	6	3	0

Parent

	A	B	C	D
A	NIL	0	1	0
B	1	NIL	1	0
C	1	2	NIL	2
D	3	0	3	NIL

(d) $k = 2$

Distance

	A	B	C	D
A	0	2	5	4
B	2	0	3	6
C	5	3	0	3
D	4	6	3	0

Parent

	A	B	C	D
A	NIL	0	1	0
B	1	NIL	1	0
C	1	2	NIL	2
D	3	0	3	NIL

(e) $k = 3$ final state

Distance

	A	B	C	D
A	0	2	5	4
B	2	0	3	6
C	5	3	0	3
D	4	6	3	0

Parent

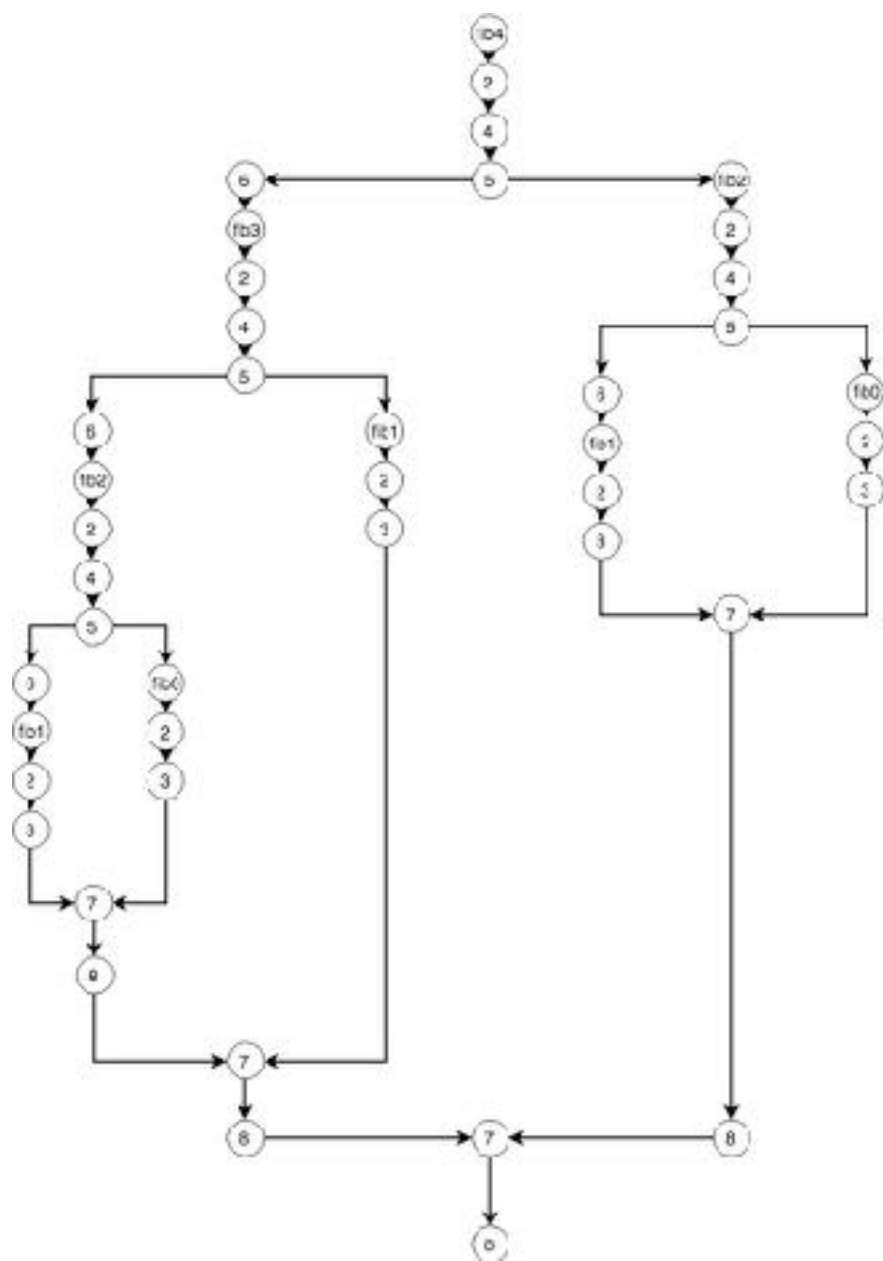
	A	B	C	D
A	NIL	0	1	0
B	1	NIL	1	0
C	1	2	NIL	2
D	3	0	3	NIL

Q3

Work = 43

span = 24

Speedup = $43/24 = 1.79$



Q4

(1)

With the atomic x, the final x value is 0. But the maximum and minimum value is not constant.

With the global variable x, the final x value would vary.

The reason is that atomic x would prevent the race condition in the problem.

Two threads would add x one by one. But the output of global variable suffers from this problem. That's to say, two threads would use the same variable at the same time. The output is related with the sequence and order of these two thread.

(2)

In non-atomic condition, the final value x is not the same. Because of the race condition, the two threads would cope with one x value at the same time. But the scheduling of these threads is undetermined. In other words, we cannot make sure the order of Produce() and Consumer(). So the final value would varies from time to time.