CS340 – Advanced Data Structures and Algorithm Design – Fall 2020
Assignment 7 – November 16, 2020

Dr. Sandra Zilles, Department of Computer Science, University of Regina

answer key

Problem 1. (4+4 marks) [The students need to give the correct table only for either (a) or (b). For the other part, it is enough if they give the solution. Subtract 1 mark for each mistake.]

(a)

	initially			after step 1			after step 2		
vertex	known	distance	previous	known	distance	previous	known	distance	previous
A		0	–	true	0	–	true	0	–
B		∞	–		5	A		5	A
C		∞	–		3	A	true	3	A
D		∞	–		∞	–		10	C
E		∞	–		∞	–		10	C
F		∞	–		∞	–		∞	–
G		∞	–		∞	–		∞	–

	after step 3			after step 4			after step 5		
vertex	known	distance	previous	known	distance	previous	known	distance	previous
A	true	0	–	true	0	–	true	0	–
B	true	5	A	true	5	A	true	5	A
C	true	3	A	true	3	A	true	3	A
D		10	C		10	C		9	E
E		8	B		7	G	true	7	G
F		∞	–		∞	–		8	E
G		6	B	true	6	B	true	6	B

	after step 6			after step 7		
vertex	known	distance	previous	known	distance	previous
A	true	0	–	true	0	–
B	true	5	A	true	5	A
C	true	3	A	true	3	A
D		9	E	true	9	E
E	true	7	G	true	7	G
F	true	8	E	true	8	E
G	true	6	B	true	6	B

results: shortest path to ...
A is [A], length 0
B is [A,B], length 5
C is [A,C], length 3
D is [A,B,E,D], length 9
E is [A,B,G,E], length 7
F is [A,B,G,E,F], length 8
G is [A,B,G], length 6.

(b)

vertex	initially dequeued	initially distance	initially previous	after step 1 dequeued	after step 1 distance	after step 1 previous	after step 2 dequeued	after step 2 distance	after step 2 previous
A		∞	–		∞	–		∞	–
B		0	–	yes	0	–	yes	0	–
C		∞	–		1	B	yes	1	B
D		∞	–		∞	–		2	C
E		∞	–		1	B		1	B
F		∞	–		∞	–		∞	–
G		∞	–		1	B		1	B
queue:	B			C,E,G			E,G,D		

vertex	after step 3 dequeued	after step 3 distance	after step 3 previous	after step 4 dequeued	after step 4 distance	after step 4 previous	after step 5 dequeued	after step 5 distance	after step 5 previous
A		∞	–		∞	–		3	D
B	yes	0	–	yes	0	–	yes	0	–
C	yes	1	B	yes	1	B	yes	1	B
D		2	C		2	C	yes	2	C
E	yes	1	B	yes	1	B	yes	1	B
F		2	E		2	E		2	E
G		2	B	yes	2	B	yes	2	B
queue:	G,D,F			D,F			F,A		

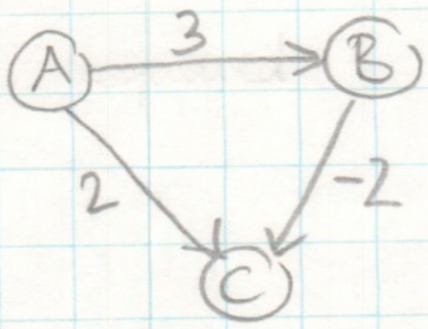
vertex	after step 6 dequeued	after step 6 distance	after step 6 previous	after step 7 dequeued	after step 7 distance	after step 7 previous
A		3	D	yes	3	D
B	yes	0	–	yes	0	–
C	yes	1	B	yes	1	B
D	yes	2	C	yes	2	C
E	yes	1	B	yes	1	B
F	yes	2	E	yes	2	E
G	yes	2	B	yes	2	B
queue:	A			empty		

results: shortest path to ...
A is [B,C,D,A], length 3
B is [B], length 0
C is [B,C], length 1
D is [B,C,D], length 2
E is [B,E], length 1
F is [B,E,F], length 2
G is [B,G], length 1.

CS340
2011/10

ASSIGNMENT 7

(2).



this graph has no cycle.

Dijkstra's alg. for source A will first set A as known with distance 0.
Then C will be marked with distance 2 and previous vertex A; B marked with distance 3 and previous vertex A. Since Dijkstra picks greedily, next it will set C known with distance 2.
C will not be updated again, since it is known. But there is a shorter path to C than [A, C], namely [A, B, C], which has length 1.

[2 marks for example, 2 marks for explanation]

(3) Since a shortest path contains at most $|V|-2$ many edges, each with weight in $\{1, 2, \dots, k\}$, the set of total possible lengths of shortest paths is $\{0, 1, 2, \dots, k\cdot(|V|-2)\} \cup \{\infty\}$.
Use an array of buckets numbered $0, 1, 2, \dots, k\cdot(|V|-2), k\cdot(|V|-2)+1$, used for storing vertices whose distance equals the bucket number ($k\cdot(|V|-2)+1$ means ∞).
→ [2 marks]

(Bucket(v) = k·(|V|−2) +1)

algorithm: (i) initially, place all vertices in bucket $k\cdot(|V|-2)+1$, except for vertex s in bucket 0. (Bucket(s) = 0) } O(|V|)

(ii) currentBucket = 0;

(iii) while currentBucket empty and currentBucket < $k\cdot(|V|-2)+1$ { currentBucket++; } } O(k·|V|) in total over time

(iv) for each v in currentBucket
{ for each $w \in V$ with $(v,w) \in E$ and Bucket(w) > currentBucket + c(v,w)
{
Bucket(w) = currentBucket + c(v,w);
w.previous = v;
}
}
currentBucket++;
goto (iii);

} O(|E|) in total over time

work on adjacency list

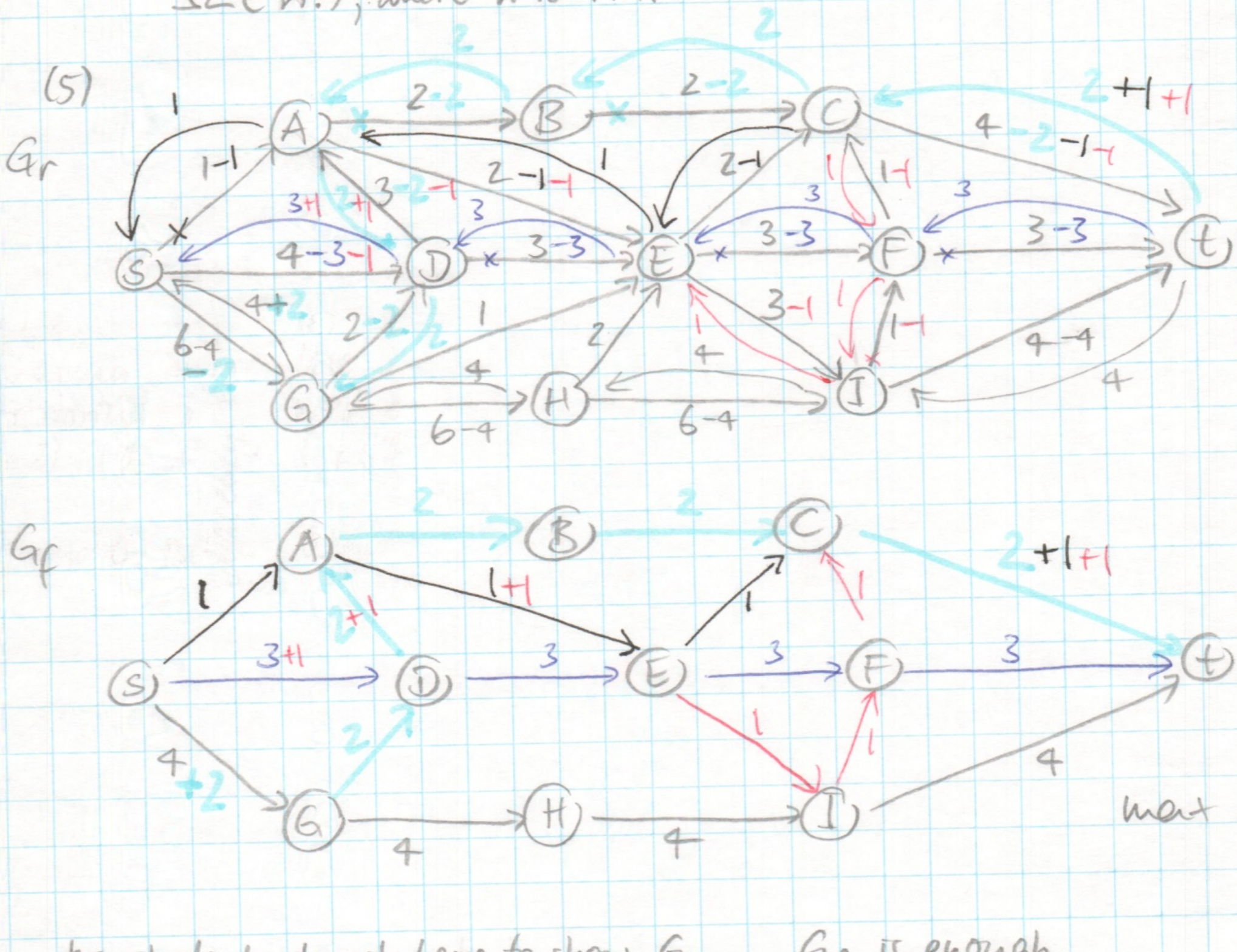
[3 marks for algorithm]
[1 mark for running time explanation]

(4) a. Since G is acyclic, G is a collection of trees.
Hence there is always at most one path from any vertex to any other vertex. [2 marks]

⇝ algorithm: BFS will work and has running time $O(|V|+|E|)$ [2 marks]

(b) In this case one would potentially have to check all possible paths from s to a vertex v. There could be exponentially many. [2 marks]
An algorithm that checks all paths would have cost of $\Omega(n!)$, where n is $|V|$. [1 mark]

(5)



the students do not have to show G_r; G_f is enough.

[give 3 marks if they find flow of 10
give 2 marks if they find flow of 7-9
give 1 mark if they find less flow, but try the right thing]