Exercise 3

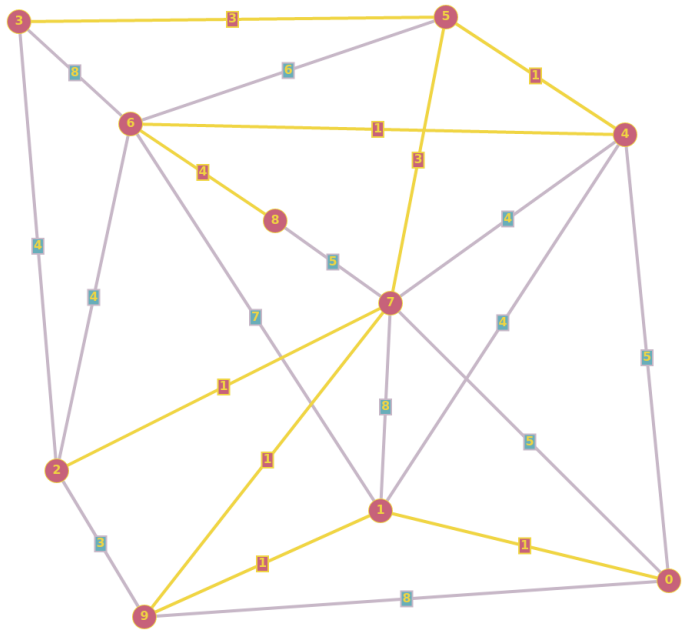
Discrete Mathematics

October 22, 2020

http://graphonline.ru/en/?graph=isCnquStAWAlFRmh

Example 21

MST with weight 16.



$F = \emptyset$

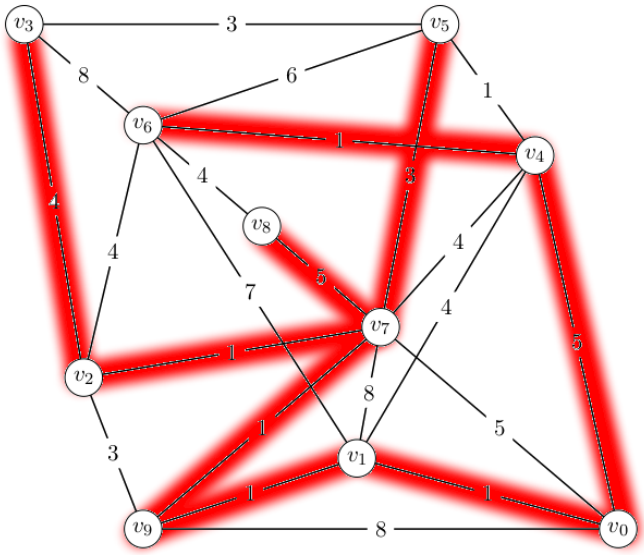
1. $F_1 = \{(0, 1, 1)\}$
2. $F_2 = F_1 \cup \{(1, 9, 1)\}$
3. $F_3 = F_2 \cup \{(9, 7, 1)\}$
4. $F_4 = F_3 \cup \{(7, 2, 1)\}$
5. $F_5 = F_4 \cup \{(4, 6, 1)\}$
6. $F_6 = F_5 \cup \{(4, 5, 1)\}$
7. $F_7 = F_6 \cup \{(5, 7, 3)\}$
8. $F_8 = F_7 \cup \{(3, 5, 3)\}$
9. $F_9 = F_8$ don't add $(2, 9, 3)$
10. $F_{10} = F_9 \cup \{(4, 7, 4)\}$
11. $F_{11} = F_{10} \cup \{(6, 8, 4)\}$

At this point we have all vertices.

Example 22

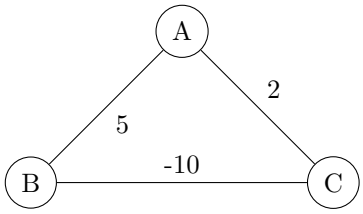
Algorithm at minute 22 in lecture

	v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	P
1	$\boxed{0}$	∞	∞	∞	∞	∞	∞	∞	∞	∞	
2		$\boxed{1}$	∞	∞	5	∞	∞	5	∞	8	v_0
3			∞	∞	5	∞	8	5	∞	$\boxed{2}$	v_1
4			5	∞	5	∞	8	$\boxed{3}$	∞		v_9
5			$\boxed{4}$	∞	5	6	8		8		v_7
6				8	$\boxed{5}$	6	6		8		v_0
6				8		$\boxed{6}$	6		8		v_7
7				8			$\boxed{6}$		8		v_4
8				$\boxed{8}$					8		v_2
9									$\boxed{8}$		v_7



Consider the graph from example 21. There the path from node 0 to node 4 has weight 7 instead of the minimum possible weight 5. This shows that the minimum spanning tree is in general not the tree with minimal distances. The distance tree that we have in this example has a total weight of 22 which is heigher than the optimum 16 that we see in example 21. Consequently, the distance tree is in general not a minimum spanning tree.

Example 23



According to steps from lecture:

	A	B	C	P	U
1	$\boxed{0}$	∞	∞		$U = \{A\}$
2		5	$\boxed{2}$	A	$U = \{A, C\}$
3		$\boxed{-8}$		C	$U = \{A, B, C\}$

The Dijkstra algorithm from the lecture failed to find the shortest path $A \to B \to C$ with weight -5 from A to C.

Example 24

Algorithm at minute 33 in lecture

S	v_0	v_1	v_2	v_3	v_4
0	0/0	∞/∞	∞/∞	∞/∞	∞/∞
1		1/1	1/1	∞/∞	∞/∞
2		1/1	1/1	2/2	2/2
3		0/3	0/3	2/2	2/2
4		0/3	0/3	1/4	1/4
		-1/5	-1/5	1/4	1/4

The algorithm returns "negative cycle". In graphs that have negative cycles, the set of shortest simple paths from one vertex to all other vertices do not necessarily form a tree.

Example 25

S	v_0	v_1	v_2	v_3	v_4	v_5	v_6
0	0/0	∞/∞	∞/∞	∞/∞	∞/∞	∞/∞	∞/∞
1		1/1	1/1	∞/∞	∞/∞	∞/∞	∞/∞
2			1/1	2/2	2/2	∞/∞	∞/∞
3			1/1		2/2	3/3	3/3
4			0/4		2/2		3/3
5					1/5		3/3
							2/6

The algorithm computes the distance correctly because it considers v_2 a second time. In contrast, Dijkstra's algorithm would look at the edge (v_5, v_2) and say "I've already visited v_2 (in terminology of the lecture $v_2 \notin V \setminus U$) and don't look at it again". Thus, v_2 gets 0 here while it would retain 1 with Dijkstra.

Example 26

Algorithm at minute 37 in lecture. http://graphonline.ru/en/?graph=aoekxjOUngh-PibWc

$i=0$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	∞	∞	∞	∞
2	∞	0	6	3	10	∞	∞
3	∞	∞	0	4	∞	9	∞
4	∞	∞	3	0	6	3	∞
5	∞	∞	∞	4	0	3	2
6	∞	∞	∞	∞	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=1$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	∞	∞	∞	∞
2	∞	0	6	3	10	∞	∞
3	∞	∞	0	4	∞	9	∞
4	∞	∞	3	0	6	3	∞
5	∞	∞	∞	4	0	3	2
6	∞	∞	∞	∞	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=2$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	7	14	∞	∞
2	∞	0	6	3	10	∞	∞
3	∞	∞	0	4	∞	9	∞
4	∞	∞	3	0	6	3	∞
5	∞	∞	∞	4	0	3	2
6	∞	∞	∞	∞	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=3$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	∞	∞	∞	∞
2	∞	0	6	3	10	15	∞
3	∞	∞	0	4	∞	9	∞
4	∞	∞	3	0	6	3	∞
5	∞	∞	∞	4	0	3	2
6	∞	∞	∞	∞	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=4$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	7	13	10	∞
2	∞	0	6	3	9	6	∞
3	∞	∞	0	4	10	7	∞
4	∞	∞	3	0	6	3	∞
5	∞	∞	7	4	0	3	2
6	∞	∞	∞	∞	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=5$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	7	13	10	15
2	∞	0	6	3	9	6	11
3	∞	∞	0	4	10	7	12
4	∞	∞	3	0	6	3	8
5	∞	∞	7	4	0	3	2
6	∞	∞	9	6	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=6$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	7	12	10	12
2	∞	0	6	3	8	6	8
3	∞	∞	0	4	9	7	9
4	∞	∞	3	0	5	3	5
5	∞	∞	7	4	0	3	2
6	∞	∞	9	6	2	0	2
7	∞	∞	∞	∞	∞	∞	0

$i=7$	$k=1$	2	3	4	5	6	7
$j=1$	0	4	5	7	12	10	12
2	∞	0	6	3	8	6	8
23	∞	∞	0	4	9	7	9
4	∞	∞	3	0	5	3	5
5	∞	∞	7	4	0	3	2
6	∞	∞	9	6	2	0	2
7	∞	∞	∞	∞	∞	∞	0

Example 27

We know two properties must hold

$$\forall e \in E : 0 \leq \phi(e) \leq w(e) \tag{1}$$

$$\forall v \in V \setminus \{s,t\} : \sum_{(v,y)\in E} \phi(v,y) = \sum_{(x,v)\in E} \phi(x,v) \tag{2}$$

As $0 \leq \gamma_1, \gamma_2 \leq 1$ we get from the first property

$$\forall e \in E : 0 \leq \gamma_1\phi_1(e) \leq \phi_1(e) \leq w(e)$$
$$\forall e \in E : 0 \leq \gamma_2\phi_2(e) \leq \phi_2(e) \leq w(e)$$

and thus

$$\forall e \in E : 0 \leq \gamma_1\phi_1(e) + \gamma_2\phi_2(e) \leq \phi_1(e) + \phi_2(e) \leq 2w(e)$$
$$\forall e \in E : 0 \leq \frac{\gamma_1\phi_1(e)}{2} + \frac{\gamma_2\phi_2(e)}{2} \leq \frac{\phi_1(e) + \phi_2(e)}{2} \leq w(e)$$

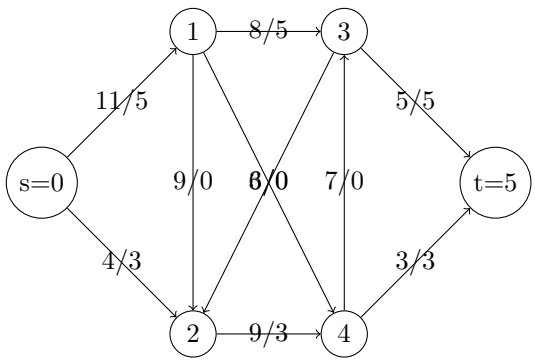
which already includes somehow$\gamma_1 + \gamma_2 \leq 1$ then ?

Example 28

1:40:00 in lecture http://graphonline.ru/en/?graph=NBkRVuUvRIyYRGAM

Path\Edge	0,1	0,2	1,2	1,3	1,4	2,4	3,2	3,5	4,3	4,5
Initial	11/0	4/0	9/0	8/0	6/0	9/0	3/0	5/0	7/0	3/0
0,2,4,5	11/0	4/3	9/0	8/0	6/0	9/3	3/0	5/0	7/0	3/3
0,1,3,5	11/5	4/3	9/0	8/5	6/0	9/3	3/0	5/5	7/0	3/3

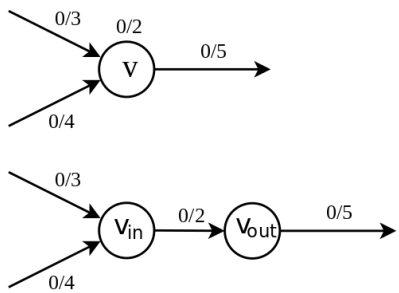
All incoming edges of the sink are saturated. Consequently, we have a maximal flow of 8.



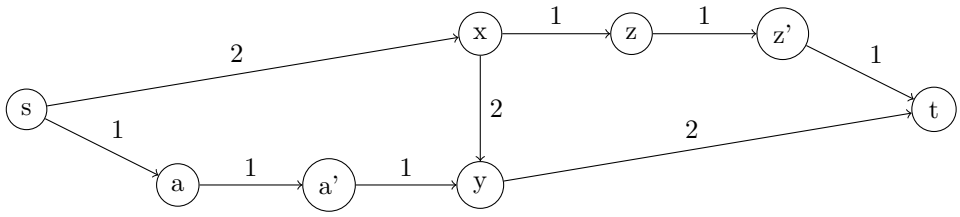
Example 29

Yes, it is possible. To do so, we transform the problem by replacing each vertex $v \in V$ by v_{in}, v_{out} where v_{in} is connected by edges going into v and v_{out} is connected to edges coming out from v and then assign capacity $c(v)$ to the edge connecting v_{in}, v_{out}. In this problem, the vertex capacity constraint is removed and thus it can be solved as regular maximum flow problem.

Example:



Example 30



Path\Edge	s,x	x,y	y,t	s,a	a,a'	a',y	x,z	z,z'	z',t
	2/0	2/0	2/0	1/0	1/0	1/0	1/0	1/0	1/0
s,x,y,t	2/2	2/2	2/2	1/0	1/0	1/0	1/0	1/0	1/0
s,a,a',y,x,z,z',t	2/2	2/1	2/2	1/1	1/1	1/1	1/1	1/1	1/1