Kent Mark

583972417

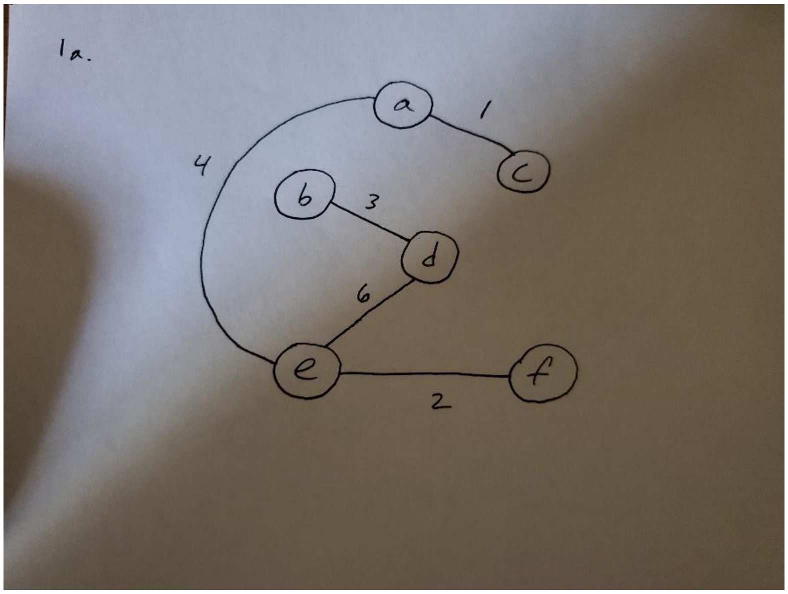
4/21/2021

Com S 311 Exam 2

Exam 2

1.

a.



b.

	1	2	3	4	5
Kruskal's	(a, c)	(e, f)	(b, d)	(a, e)	(d, e)
Prim's	(a, c)	(a, e)	(e, f)	(d, e)	(b, d)

c. I could show that G has a unique minimum spanning tree by calculating G's minimum cost.
Minimum cost = 1 + 2 + 3 + 4 + 6 = 16

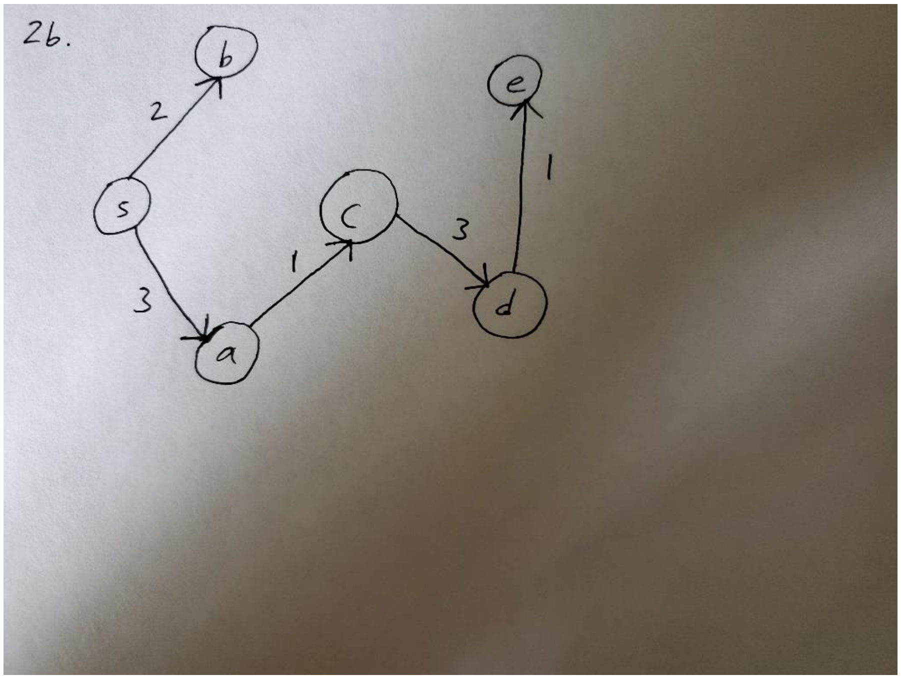
2.

a.

iter.	d[]						Selected Nodes
	s	a	b	c	d	e	
0	0	∞	∞	∞	∞	∞	s
1	0	3	2	6	∞	∞	sb
2	0	3	2	5	∞	11	sba

3	0	3	2	4	10	11	sbac
4	0	3	2	4	7	11	sbacd
5	0	3	2	4	7	8	sbacde
6	—	—	—	—	—	—	sbacde

b.



c. I do not know how to solve this question.

3. To prove that L is in P we have to build an algorithm that can make decisions in polynomial time that accepts all string in L but rejects all strings not in L. This vertex algorithm will work where a user inputs: a binary string X_n where n is the length.

   Pseudo code:

   i) y ← Reverse x o(n)
   ii) z ←compliment each bit of y(on)
   iii) if y accepts z in polynomial time, then the algorithm/program outputs "YES"
   iv) else y rejects z in polynomial time, output "NO"

   Correctness Check:

   Repeated complementation and reversal of a string will return the string in L. Therefore if y accepts z or rejects z each of the steps will run in polynomial time. As such, L is in P.

4. This algorithm was written in C!

```
// C program to count string num

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

//k is the total num of characters

int strCount(int k, int b, int c){
//base case

if(b < 0 || c < 0){
return 0;
}

if(k == 0) {
return 1;
}

if(b == 0 && c == 0){
return 1;
}

//in these three cases we will go with the first, second, or third choice, k decreases by 1

int alg = strCount(n – 1, b, c);
alg += strCount(n – 1, b – 1, c – 1);
alg += strCount(n – 1, b, c);

return alg;
}
```

5. I do not know how to solve this problem.