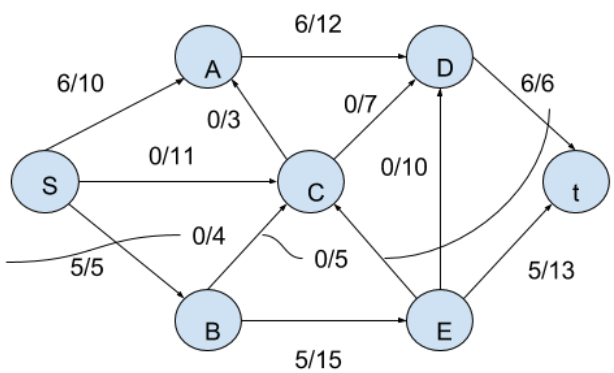
CS 4310 - Design & Analysis of Algorithms
Network Flow, NP, B-Tree Assignment
4/13/2017
Mariia Kravtsova

1. The vertices included in the min cut are: source
2. The maximum flow is 11, see the graph below for mincut



3. It is not specified where does the problem L belong to as in P, NP, NP Complete or NP Hard, it only says that it is reducible to NP Complete. Hence, I think that the professor has not proven that P = NP. To prove P = NP we could solve an NP-complete problem in polynomial time, we could solve every problem in NP in polynomial time.
4.

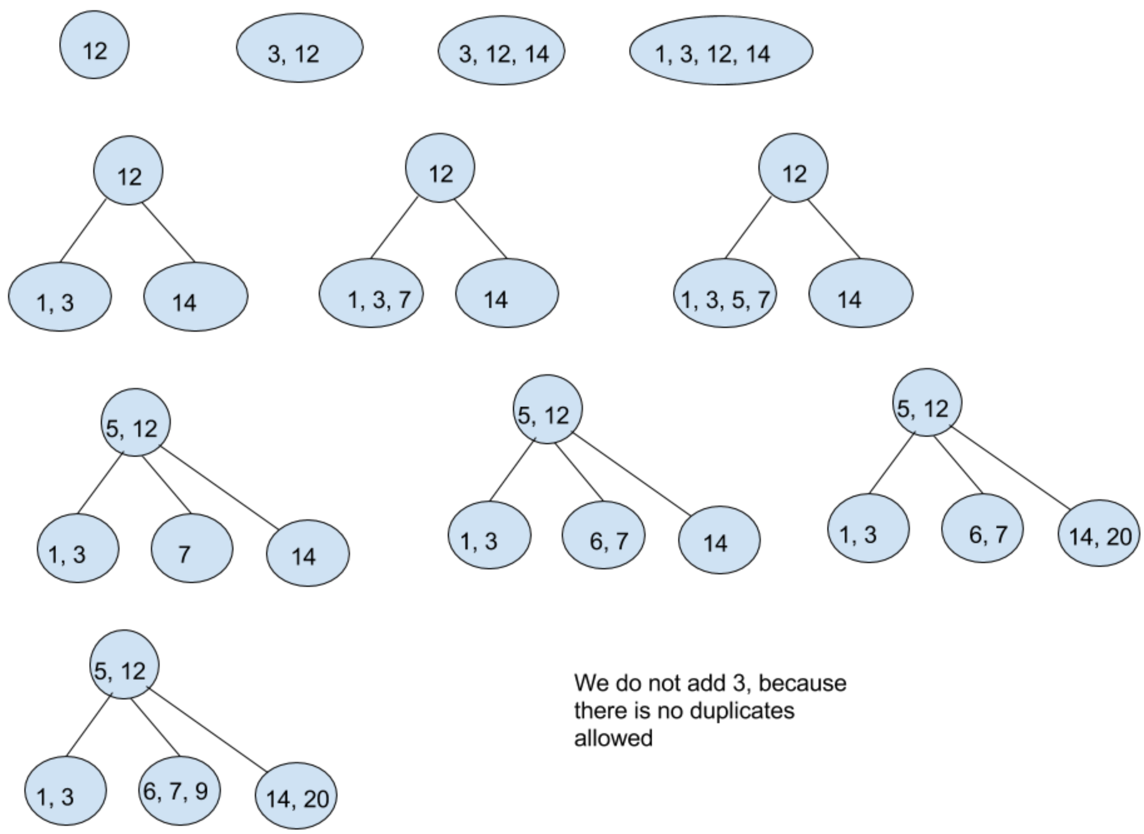
a	b	c	b + c	a <-> (b+c)
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	0	0
0	1	1	1	0
0	1	0	1	0
0	0	1	1	0
0	0	0	0	1

$\neg B = a \cdot \neg b \cdot \neg c + \neg a \cdot b \cdot c + \neg a \cdot b \cdot \neg c + \neg a \cdot \neg b \cdot c$

$B = (\neg a + b + c) \cdot (a + \neg b + \neg c) \cdot (a + \neg b + c) \cdot (a + b + \neg c)$

5. Take an input n, then draw a truth table for your problem S. Write the DNF by using ands for the variables for example ($a \wedge b \wedge c$), then each of that instance should be or-ed. For S to be satisfiable we will need to find the first instance to equal to 1, and in this case the algorithm will run in polynomial time O(n).

6.



We do not add 3, because there is no duplicates allowed

7.