Jackie Diep
CSC 413 Algorithms
Assignment #4
4/11/2021

-Chapter 15
1. [10 pts] Page 396:15.4-1

15.4-1

	Y	0	1	0	1	1	0	1	1	0
X	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1
0	0	1	1	2	2	2	2	2	2	2
0	0	1	1	2	2	2	3	3	3	3
1	0	1	2	2	3	3	3	4	4	4
0	0	1	2	3	3	3	4	4	4	5
1	0	1	2	3	4	4	4	5	5	5
0	0	1	2	3	4	4	5	5	5	6
1	0	1	2	3	4	5	5	6	6	6

LCS: <0,1,0,1,0,1>

2. [20 pts] Dynamic Programming: Determine a Longest Common Subsequence (LCS) for the following two strings using dynamic programming approach. You need to illustrate the step-by-step procedure based on a table.

'ncaa tournament' and 'north carolina'

```
if (i == 0 && j == 0)
{        LCS[i][j] = 0;              }
else if(x[i] == y[j])
{        LCS[i][j] = 1 + LCS[i-1][j-1];              }
else
{        LCS[i][j] = max(LCS[i-1][j]l, LCS[i][j-1];         }
```

Ch. 15 #2

	y	n	o	r	t	h	-	c	a	r	o	l	i	n	a
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
c	0	1	1	1	1	1	1	2	2	2	2	2	2	2	2
a	0	1	1	1	1	1	1	2	3	3	3	3	3	3	3
a	0	1	1	1	1	1	1	2	3	3	3	3	3	3	4
-	0	1	1	1	1	1	2	2	3	3	3	3	3	3	4
t	0	1	1	1	2	2	2	2	3	3	3	3	3	3	4
o	0	1	2	2	2	2	2	2	3	3	4	4	4	4	4
u	0	1	2	2	2	2	2	2	3	3	4	4	4	4	4
r	0	1	2	3	3	3	3	3	3	4	4	4	4	4	4
n	0	1	2	3	3	3	3	3	3	4	4	4	4	5	5
a	0	1	2	3	3	3	3	3	4	4	4	4	4	5	6
m	0	1	2	3	3	3	3	3	4	4	4	4	4	5	6
e	0	1	2	3	3	3	3	3	4	4	4	4	4	5	6
n	0	1	2	3	3	3	3	3	4	4	4	4	4	5	6
t	0	1	2	3	3	4	4	4	4	4	4	4	4	5	6

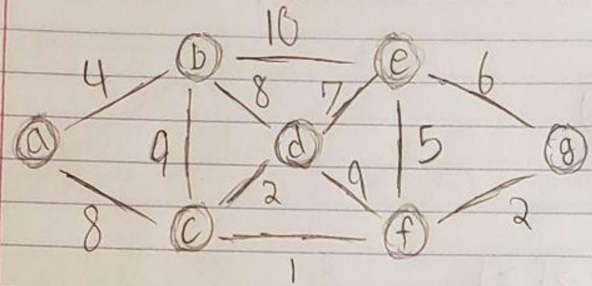
LCS: <n, c, a, o, n, a>

-Chapter 23

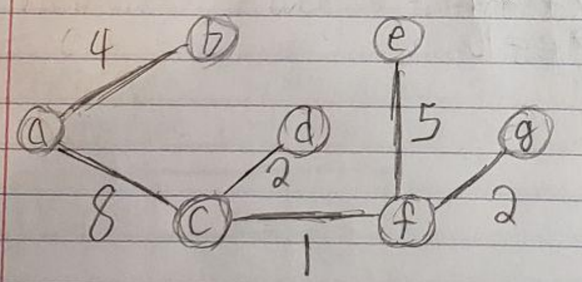
1. [25 pts] Finding the Minimum Spanning Trees (MST) of following graphs.

(a) Prim's algorithm

Prim's MST

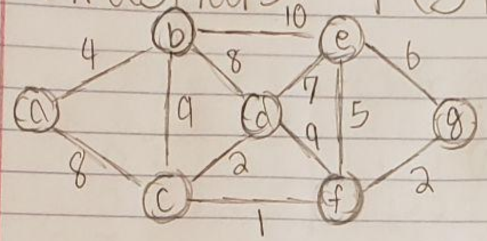


a	Choose "a" as root,
4→b	a→b has least weight b/t 4 and 8
a→c	Next, b→d and a→c have the same weight, so either may be chosen. I will add a→c.
c→f	c→f is next least weight b/t 1, 2, 8, 9
c→d/f→g	Next, c→d and f→g have the same weight, but both go to diffarent nodes and are the lowest remaining paths. As such, I will add both
f→e	f→e is the next least weight b/t 5, 8, 9



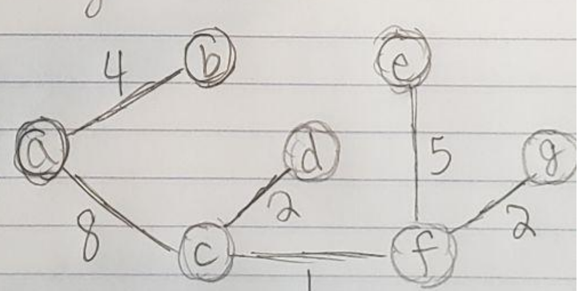
(b) Kruskal's algorithm

Kruskal's MST



Organizing by weights: c→f = 1
c→d = 2
f→g = 2
a→b = 4
e→f = 5
e→g = 6
d→e = 7
a→c = 8
b→d = 8
b→c = 9
d→f = 9
b→e = 10

Add c→f
Add c→d
Add f→g
Add a→b
Add e→f
Skip e→g as both are already added
Skip d→e
Add a→c
Skip b→d
Skip b→c
Skip d→f
Skip b→e



(c) Compare and comment on your results obtained in (a) and (b)

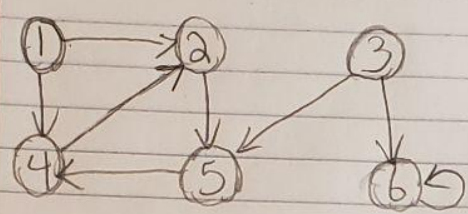
Both algorithms provide the same Minimum Spanning Trees. Differences could come from selecting a different option between b → d or a → c as both have the same cost, but that is not due to the algorithms.

-Chapter 22
1. P592-593: 22.1-2 (10 pts): Give an adjacency-list representation for a complete binary tree on 7 vertices. Give an equivalent adjacency-matrix representation. Assume that vertices are numbered
from 1 to 7 as in a binary heap.

0	1	2	3	4	5	6	7
1	0	1	1	0	0	0	0
2	0	0	0	1	1	0	0
3	0	0	0	0	0	1	1
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0

2. P601-602: 22.2-1 (10 pts): Show the d and π values that result from running breadth-first search on the directed graph of Figure 22.2(a), using vertex 3 as the source.

22.2-1



	vertex	π	d	
Start:	3	NIL	0	Queue = 3
	5	3	1	Queue = 5, 6
	6	3	1	
	4	5	2	Queue = 6, 4

Remove 6 from queue as it loops to a visited node (itself)

2	4	3	Queue = 2

Remove 2 from queue as it returns to a visited node (5)
1 remains undefined as there is no path to it

FINAL

Vertex	π	d
1	NIL	∞
2	4	3
3	NIL	0
4	5	2
5	3	1
6	3	1