CMPE 130 Midterm Exam #2 Fall 2015

15:00—16:15 Tuesday, November 3, 2015

42

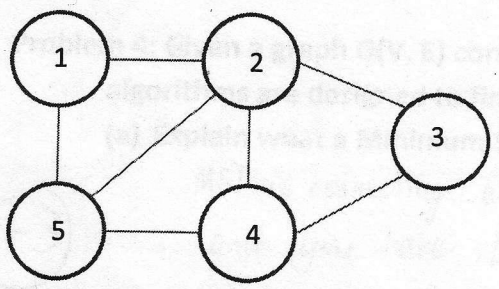
Student Name _____(print)

Student ID_ _____

75

Problem 1: For the set of {1,4, 5, 10, 16, 17, 21} of keys, draw binary search trees of height 2, 3, 4, 5 and 6. (10 points) see extra paper

Problem 2: Write the adjacency-matrix representation of Graph (a) and Graph (b) below. (10 points)



Graph (a)

a.

1 → 2 → 5

2 → 1 → 3 → 4 → 5

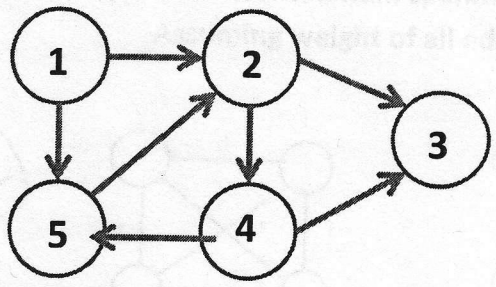
3 → 2 → 4

4 → 2 → 3 → 5

5 → 1 → 2 → 4

	1	2	3	4	5
1	0	1	0	0	1
2	1	0	1	1	1
3	0	1	0	1	0
4	0	1	1	0	1
5	1	1	0	1	0

✓



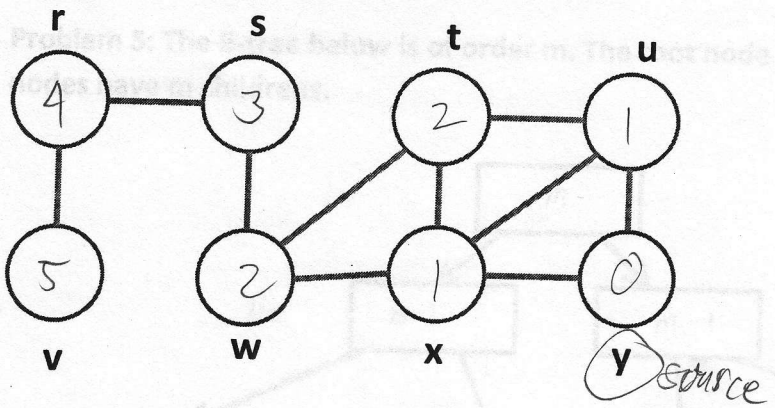
Graph (b)

b.

	1	2	3	4	5
1	0	1	0	0	1
2	0	0	1	1	0
3	0	0	0	0	0
4	0	0	1	0	1
5	0	1	0	0	0

✓

Problem 3: Show the step-by-step result of running BFS on the graph below using y as the source. (20 points) see extra paper queue



✓

Problem 4: Given a graph G(V, E) consisting of a set of vertices V and a set of edges E, algorithms are designed to find Minimum Spanning Tree (MST).

(a) Explain what a Minimum Spanning Tree is. (10 points)

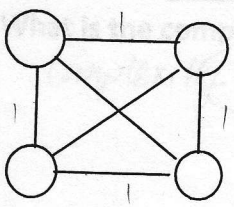
MST is connecting all nodes with out any loops and you have $E = V - 1$

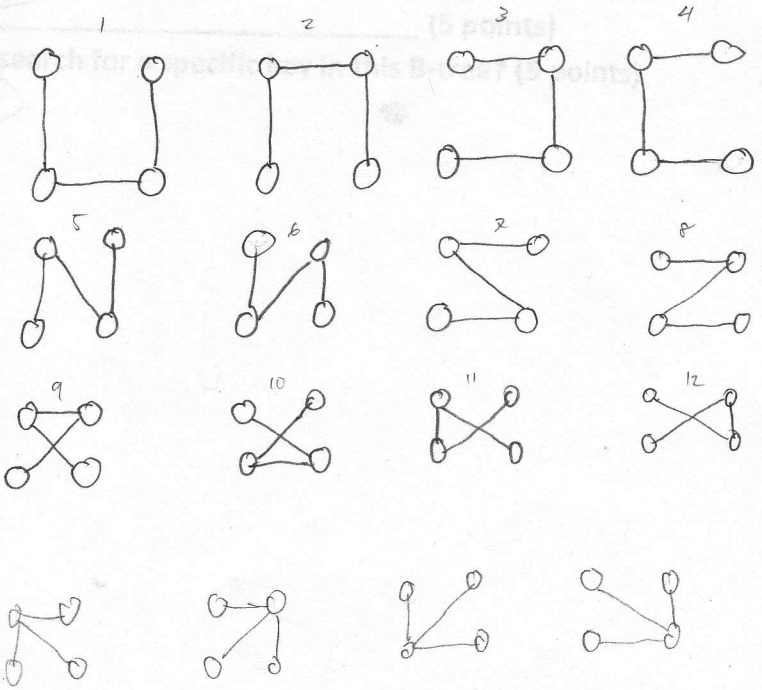
- all
- undirected

-3

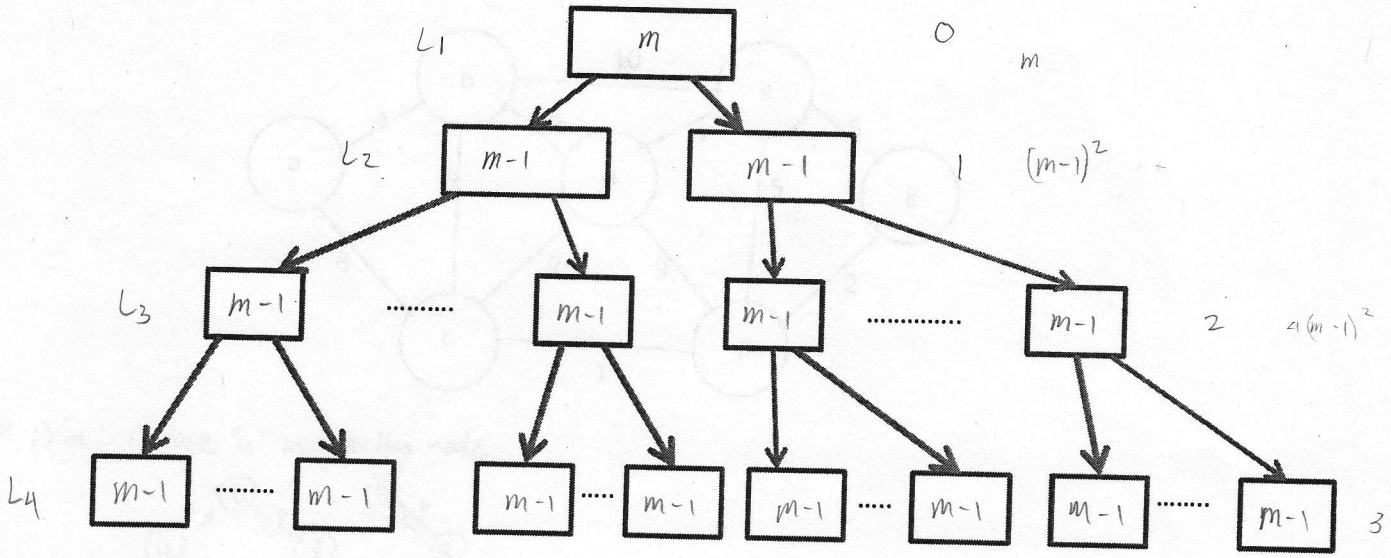
(b) Draw all minimum spanning trees of the graph below
Assuming weight of all edges is "1" (10 points)

-2





Problem 5: The B-tree below is of order m. The root node has two children and the non-leaf nodes have m childrens.



(a) How many nodes are there at level 4? = $8(m-1)^2$ (5 points)

[Note: root is level 1 while the leaves are at level 4]

(b) How many keys are stored at level 4? = $8(m-1)^2$ (5 points)

-20

(c) If m=1000, how many keys can we store at level 4?

Answer= $(7992)^2$ (5 points)

$8(1000-1) = 8(999)$

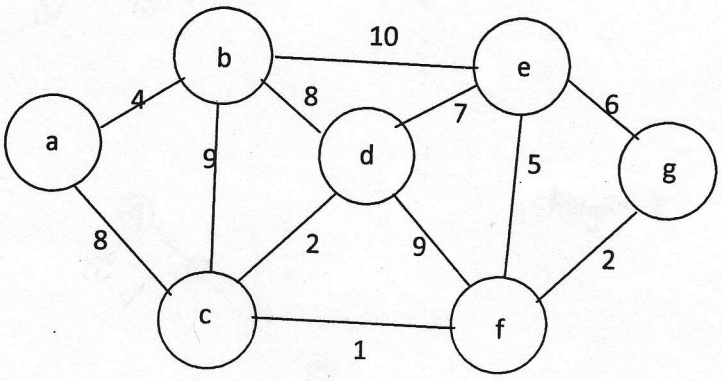
(c) What is the complexity to search for a specific key in this B-tree? (5 points)

complexity $O(V^2)$

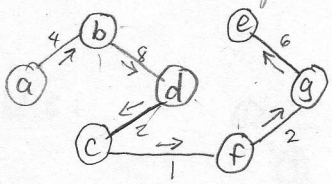
Problem 6: Use the graph below to find a Minimum Spanning Tree by

(a) Prim's Algorithm (10 points) and start anywhere

(b) Kruskal Algorithm (10 points) start from least

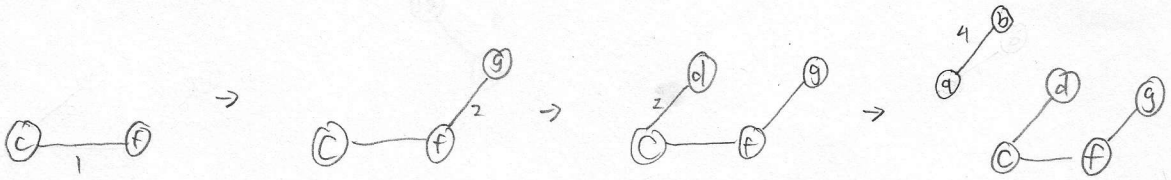


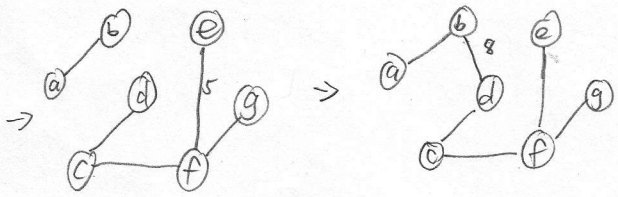
a. Prim: Choose "a" as starting node



b. Kruskal:

path	c-f	f-g	c-d	a-b	e-f	e-g	d-e	b-d	b-c	b-e
weight	1	2	2	4	5	6	7	8	9	10





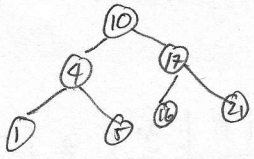
✓

Problem 1:

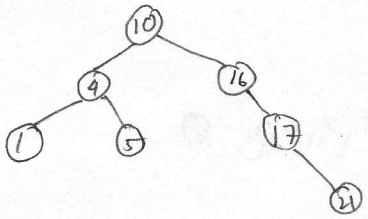
set {1, 4, 5, 10, 16, 17, 21}

height: 2, 3, 4, 5, 6

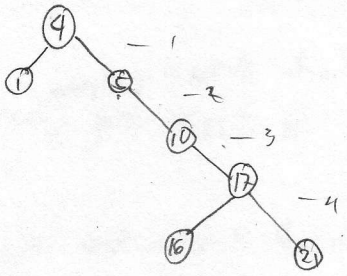
height 2:



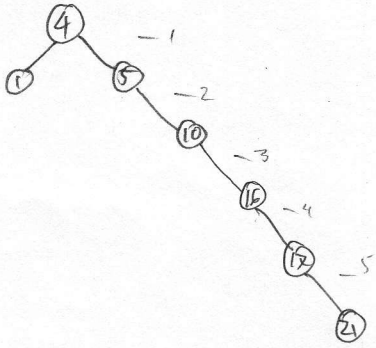
height 3:



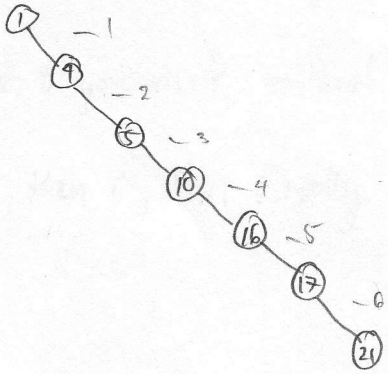
height 4:



height 5:

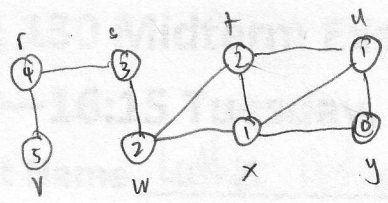


height 6:



✓

problem 3:



	r	s	t	u	v	w	x	y
d	4	3	2	1	5	2	1	0
π	v	w	u	y	r	t/x	y	source

Q: y, u, x, t, w, s, r, v

take y as source, everything is set to infinity then you visit u and add it to Queue

you then visit x and add to Queue

then visit t which is 2 hops away add to Quee

From either x or t you visit w and add it to Quee

then visit s, then r, then finally v