CS 300 Data Structures
Problem Set # 23 - Review

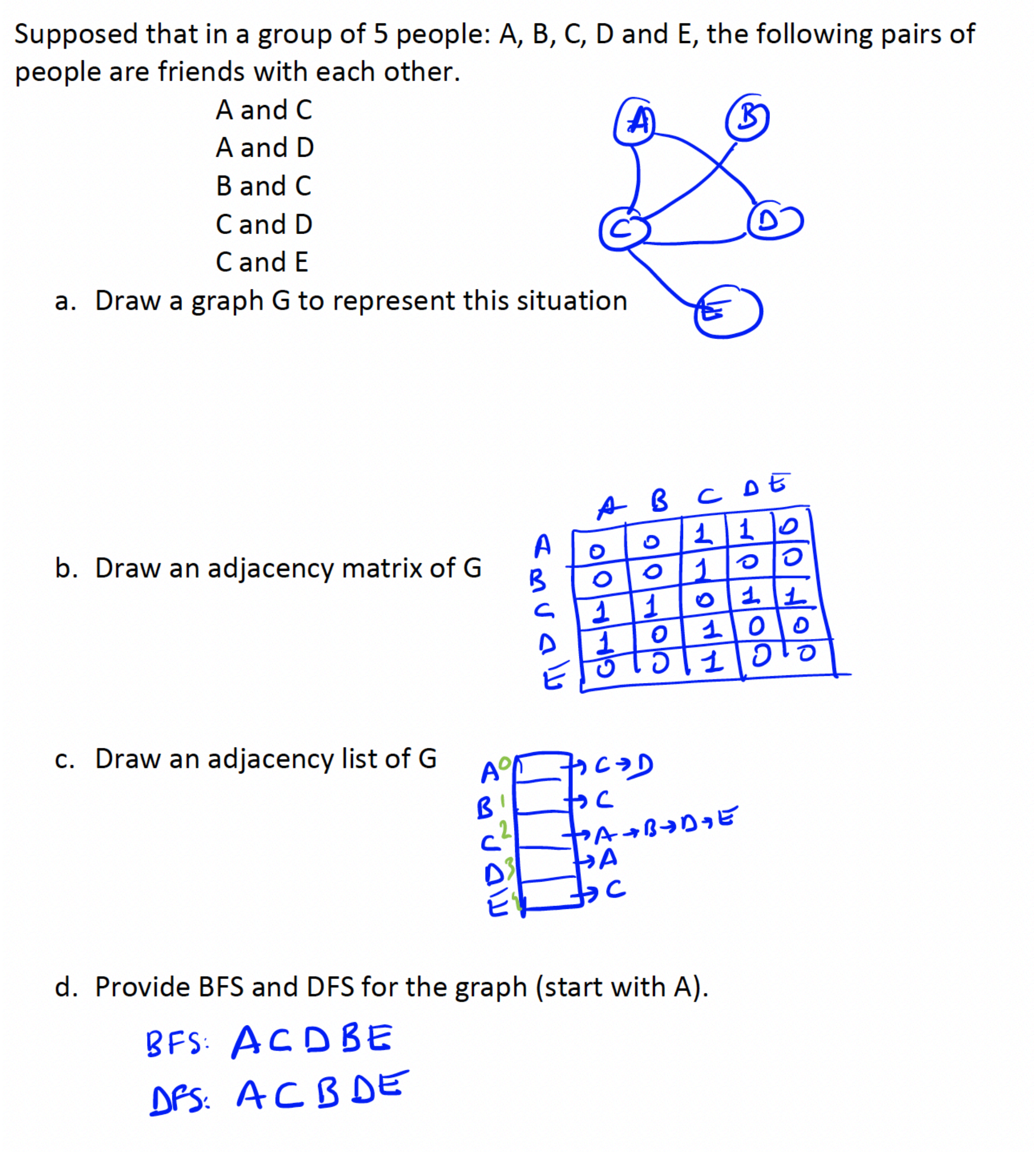
1. What graph traversal algorithm uses a queue to keep track of vertices which need to be processed?

BFS

2. Which graph representation allows the most efficient determination of the existence of a particular edge in a graph?

Adjacency Matrix

3.



Supposed that in a group of 5 people: A, B, C, D and E, the following pairs of people are friends with each other.

A and C
A and D
B and C
C and D
C and E

a. Draw a graph G to represent this situation

b. Draw an adjacency matrix of G

	A	B	C	D	E
A	0	0	1	1	0
B	0	0	1	0	0
C	1	1	0	1	1
D	1	0	1	0	0
E	0	0	1	0	0

c. Draw an adjacency list of G

A 0 → C → D
B 1 → C
C 2 → A → B → D → E
D 3 → A
E 4 → C

d. Provide BFS and DFS for the graph (start with A).

BFS: A C D B E
DFS: A C B D E

3. Choose the data structure

   a. You are asked to implement a database for a telephone company. The database will contain the names and phone numbers of all its clients. For the following operation, the company wants the average running time to be small as possible: 1) insert a new entry, 2) search for a person's phone numbers, and 3) remove an entry from the database. What data structure will you use?

   Hashing

   b. You are asked to develop a scheduler of an operating systems. The scheduler is supposed to pick the shortest job first, which data structure would you choose to implement the schedule?

   Priority Queue

   c. If you are asked to implement the same schedule to pick the first come first bases, which data structure would you choose?

   Queue

4. We want to implement an Abstract Data Type (ADT) with operations Insert, ReportMin, GetMax, and Contains. The operation ReportMin reports the min without deleting it, while GetMax both reports the max and deletes it from the ADT. The operation Contains(x) returns a boolean value indicating whether x is in the ADT or not.

   For each of the following requirements, describe a data structure (either a standard data structure or a modification of a standard data structure or a data structure of your own design) that implements the ADT and satisfies the requirements. Your description should be brief, but be sure to give enough description that we can tell what you have in mind. The value n represents the number of items in the data structure.

   a. Insert, ReportMin, Contains, and GetMax each take time O(log n) in the worst case.

   Balanced BST

   b. Insert and ReportMin each take time O(1) in the worst case.

   LinkedList + a variable to store current minimum

   c. Insert, Contains, and ReportMin each take expected time O(1).

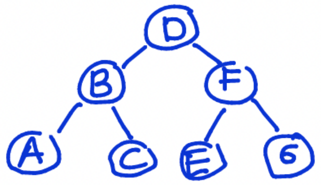
   Hash Table + a variable to store current minimum

   d. GetMax takes time O(1) in the worst-case and Contains takes time O(log n) in the worst case.
   Sorted Array

5. Insert the following sequence of numbers 23, 46, 12, 21, 75, 5, 3 into a hash table of size 9 using h(x) = x%9 as a hash function. Use a linear probing for resolving collisions.

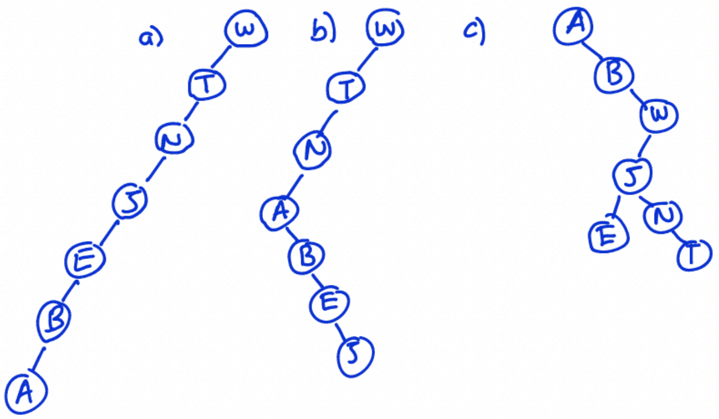
0	
1	46
2	12
3	21
4	75
5	23
6	5
7	3
8	

6. The height of a Binary Search Tree (BST) depends on the order in which the keys are inserted into an empty tree if no balancing operation is performed. Given an initially empty BST, in what order will you insert the keys A, B, C, D, E, F, G so that the height of the BST is a minimum. Use the natural alphabetic order to sort the keys, i.e., A < B < C < D < E < F < G.



7. Beginning with an empty binary search tree, what binary search tree is formed when you insert the following values in the order given?

   a) W,T,N,J,E,B,A
   b) W,T,N,A,B,E,J
   c) A,B,W,J,N,T,E



8. Both hash tables and binary search trees are used for storing key value pairs. Hash tables promise search, insert and delete operations in constant time,

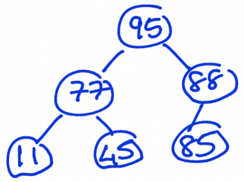
whereas binary search trees need O(log n) time for the same operations. Why would anyone use binary search trees instead of hash tables?

BST uses memory more efficiently
Rehashing is costly
Finding max, min is const in hashing

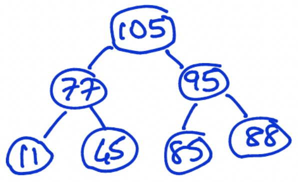
9. The following array represents a heap.

0	unused
1	95
2	77
3	88
4	11
5	45
6	85

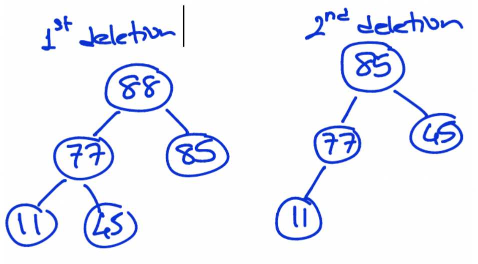
a. Show the tree represented by this array.



b. Show the contents of the array after 105 is inserted into the original heap



c. Show the contents of the array after `detete()` is executed twice on the original heap



10. (5 pts.) Given a circular array-based queue capable of holding 100 objects. Suppose the queue is initially empty, and then objects are put into the queue at the rate of 10 per minute while meantime they are processed and removed from the queue at the rate of 5 per minute. After 120 elements have been added to the queue, which of the following is true?

    a. You can't add 120 elements to an array holding 100 entries.
    b. There will be 60 elements in the queue, 20 of them at the front of the array where the queue started, and 40 at the other end.
    c. There will be 60 elements in the queue, 30 of them at the front of the array where the queue started, and 30 at the other end.
    d. There will be 60 elements in the queue, 40 of them at the front of the array where the queue started, and 20 at the other end.