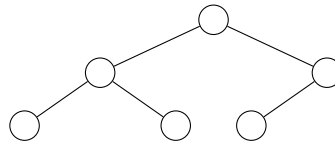


**Short Time Questions 25 - points**

1. (5 points) In the following BST, which node is the largest? you can circle it



2. (5 points) True or False:

- (a)  $2n^2 + 1 = O(n^2)$
- (b)  $\sqrt{n} = O(\log n)$
- (c)  $\log n = O(\sqrt{n})$
- (d)  $3n^2 + \sqrt{n} = O(n^2)$
- (e)  $\log n = O(n^{-1/2})$

3. (5 points) Describe the advantages and disadvantages of linear probing and chaining to solve collisions in hash tables.

4. (5 points) Can Prim's and Kruskal's algorithms yield different minimum spanning trees? Explain why or why not.?

5. (5 points) Quick sort or Bubble sort?

1. (10 points) What algorithm would you use to detect cycles in graphs? Depth First Search or Breadth First Search? Why did you choose this algorithm? List the steps clearly in bullet points that will lead to cycle detection using the algorithm you choose.
2. (10 points) A mother vertex in a directed graph  $G = (V, E)$  is a vertex  $v$  such that all other vertices  $G$  can be reached by a directed path from  $v$ . Give an  $O(n+m)$  algorithm to test whether a given vertex  $v$  is a mother of  $G$ , where  $n = |V|$  and  $m = |E|$ .

3. (10 points) Using Big-O, determine what is the worst-case scenario for the following data structure and operations (Access and Searching are Different):

	Access	Search	Insert	Deletion	Find Max
Binary Search Tree					
AVL Tree					
Hash Table					
Array	O(1)	O(n)			
Stack				O(1)	
Max Heap					

Table 1: Common Data Structure Operaions

**Long Time Questions** 45 - points

1. (15 points) Suppose an array  $A$  consists of  $n$  elements, each of which is red, or blue. We seek to sort the elements so that all **the reds come before all the blue**, The only operation permitted on the keys are
  - $\text{Examine}(A,i)$  – report the color of the  $i$ th element of  $A$ .
  - $\text{Swap}(A,i,j)$  – swap the  $i$ th element of  $A$  with the  $j$ th element.

Find a correct and efficient algorithm for red-blue sorting. (or explain in English, you can draw the array to illustrate your idea) **There is a linear-time solution.**

2. (15 points) Find the Kth largest element in a number stream. Design a class or pseudocode or use English to efficiently find the Kth largest element in a stream of numbers. The class should have the following two properties:

1. The constructor of the class should accept an integer array containing initial numbers from the stream and an integer K.
2. The class should expose a function `add(int num)` which will store the given number and return the Kth largest number.

*What is the Time Complexity of finding the kth largest number? and what is the Space Complexity (what would you keep in memory)?*

#### Example

Given the input array  $\mathbf{A} = [4, 1, 3, 12, 7, 14]$  and  $K = 3$ , consider the following operations:

- (a) Calling `add(6)` should return 7.
- (b) Calling `add(13)` should return 12.
- (c) Calling `add(4)` should still return 12.

3. (15 points) Given a list of integers: [15, 22, 13, 27, 12, 10, 20, 25], use the Randomized Quick Sort algorithm to sort the array. In Randomized Quick Sort, the pivot is selected randomly at each step instead of using a fixed strategy like choosing the first, middle, or last element as the pivot. (Show Recursive Tree)