Name: Qadeer Hussain

Class:BS

Department: Computer Science

Section:C

Roll No: 14

Semester: 4th

Subject: Design and Analysis of Algorithms

QUESTION NO:1

1. For each of the following problems answer True or False and BRIEFLY JUSTIFY you answer. (2 Marks)

Suppose O(1) time complexity, then the runtime of the code is exactly the same for all possible inputs.

Ans: False,because O(1) An algorithm is said to run in constant time if it requires the same amount of time regardless of the input size.

a) O(n^2 log n) is faster than O(n^2.1).

Ans: True,Because O(n^2 log n) is much closer to O(n) than to O(n^2.1) .

QUESTION NO:2

2. List the following functions by increasing growth rate. If two functions have the same growth rate, state that fact. No justification is needed. (2 Marks)

O(log n) O(1.1n) O(n log n) O(n (log n)^2) O(3 log n) O(2^5) O(n^34)

Ans:O(2^5) O(log n) O(n log n) O(3 log n) O(n(log n)^2) O(n^34) O(1.1n)

QUESTION NO:3

3. What is the time complexity of following code: (4 Marks each)

a) int function(int n)

```
{
  int c = 0;
  for (i = 0; i <=n; i++)
  for (j = i; j>=0; j--)
  c = c + 1;
  return c;
  }
```

Ans:O(n*n)

b) int i, j, k = 0;

```
  for (i = n / 2; i <= n; i++)
{
```

```
  for (j = 2; j <= n; j = j * 2)
  {
  k = k + n / 2;
  }
  }
```

Ans: O(n log n)

c)

```
void function(int n)
{
int c = 0;
for (i=n/2; i<=n; i++)
for (int j=1; j<=n; j = 2 * j)
  for (int k=1; k<=n; k = k * 2)
c++;
  }
```

Ans:O(n log n)

d)

```
  void function(int n)
  {
  int c = 0;
  for (i=n/2; i<=n; i++)
  for (j=1; j+n/2<=n; j = j++)
for (k=1; k<=n; k = k * 2)
  c++;
  }
```

Ans: O(n^2 log n)

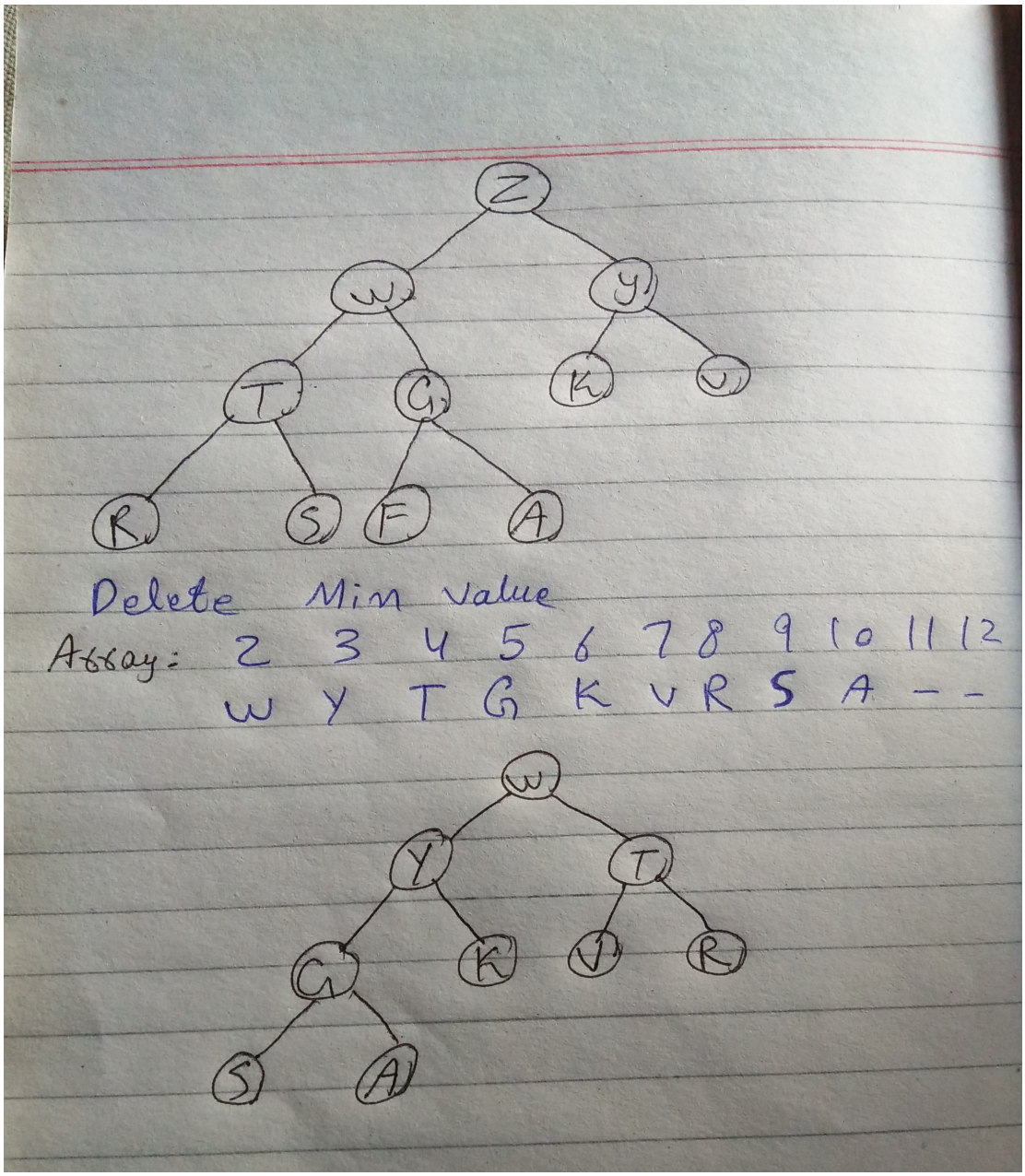
QUESTION NO:4

4. Create a Min heap and then delete the Min value from it. Show your result in the form of tree and array.

0 1 2 3 4 5 6 7 8 9 10 11 12 13

- Z W Y T G K V R S F A - -

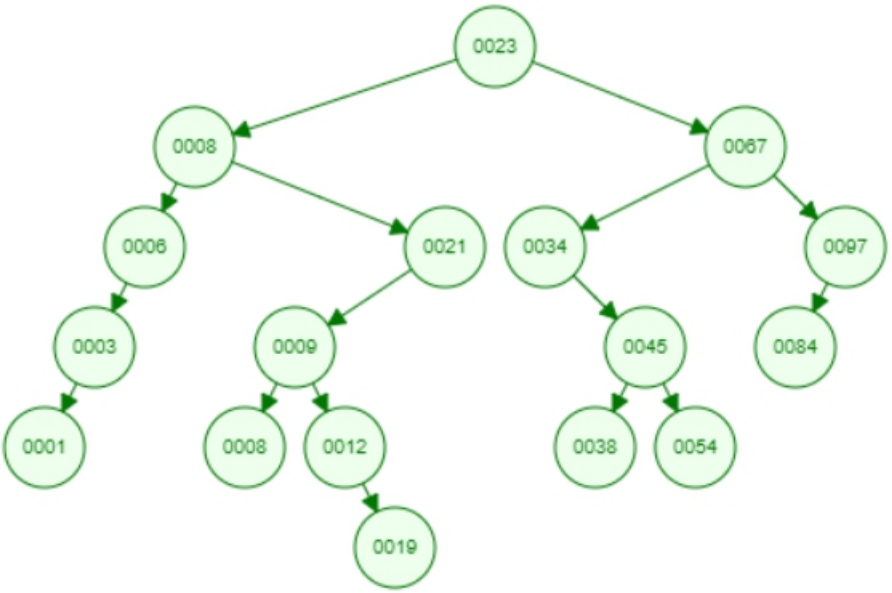
Ans:



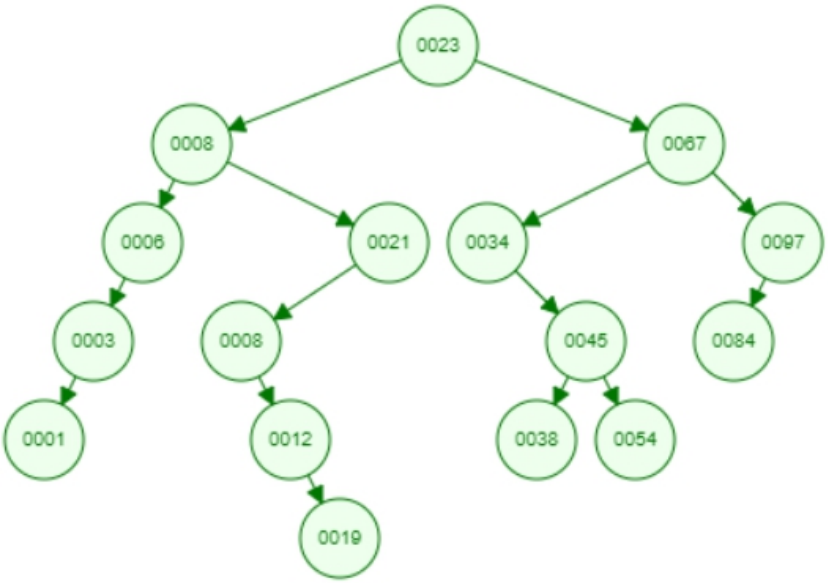
QUESTION NO:5

5. Create a binary SEARCH tree and apply these operations:

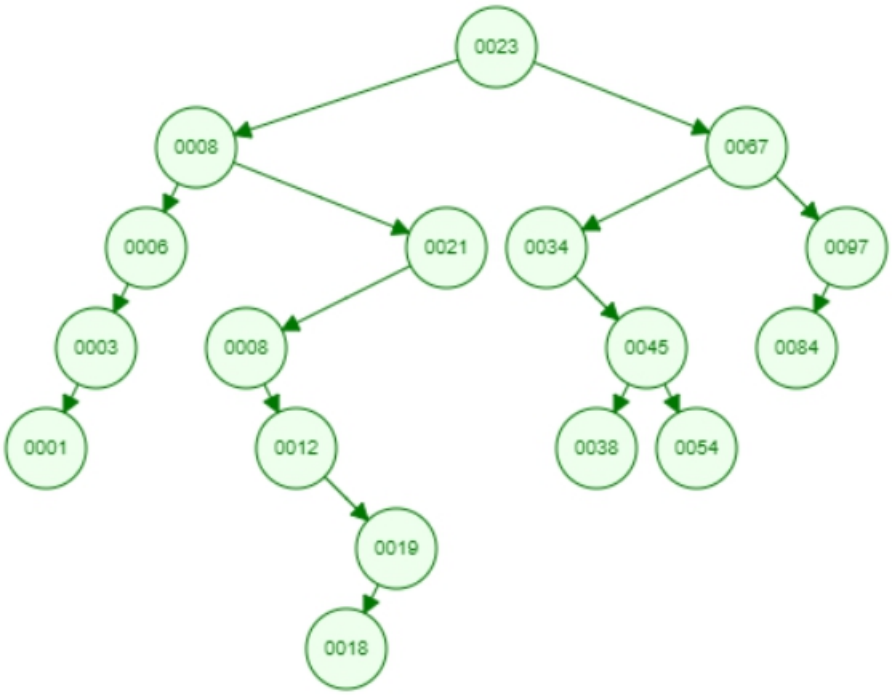
Values are 23, 67,8,97,21,34,6,3,1,9,45,8,38,54,12,19,84



a) Delete 9



b) Add 18


0023
0008
0067
0006
0021
0034
0097
0003
0008
0045
0084
0001
0012
0038
0054
0019
0018