Question (1)

a) $ceil(\ log_2(23+1)\ ) - 1 = 4$, O(log n)
b) Large "not sure"
c) n/2 "not sure"
d) Better
e) n(n-1) = 8 * 7 = 56
f) inorder
g) all answers are correct :(
h) close
i) array

Question (2)

a) False
   in a binary search tree …
   or …less than the value of its parent
b) True
c) False
   …is always stored at the left most leaf of the tree
d) False
   …faster … in case of big input size
e) True
f) False
   postorder

Question (3)

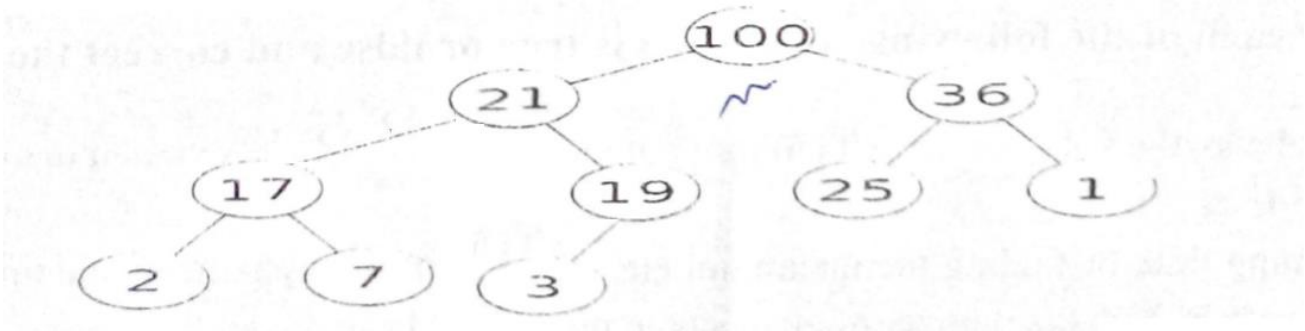
a) 16
b) ((5+3)/(2))*((2+8)-(3*2)) "inorder"
   53+2/28+32*-* "postorder"
c)
```
Class TreeNode{
public:
char value;
TreeNode* L,R;

TreeNode()
{
value=0;
L=R=NULL;
}
};
```
d)
```
void preorder(TreeNode *node)
{
if(node==NULL)return;
cout<< node ->value  << " ";
preorder(node->L);
preorder(node->R);
}
```
e)
```
void terminal_count(TreeNode *node, int & count)
{
if(node==NULL)return;
if(node->L == NULL && node->R == NULL)count++;
terminal_count (node->L, count);
terminal_count (node->R, count);
}
```

Question (4)

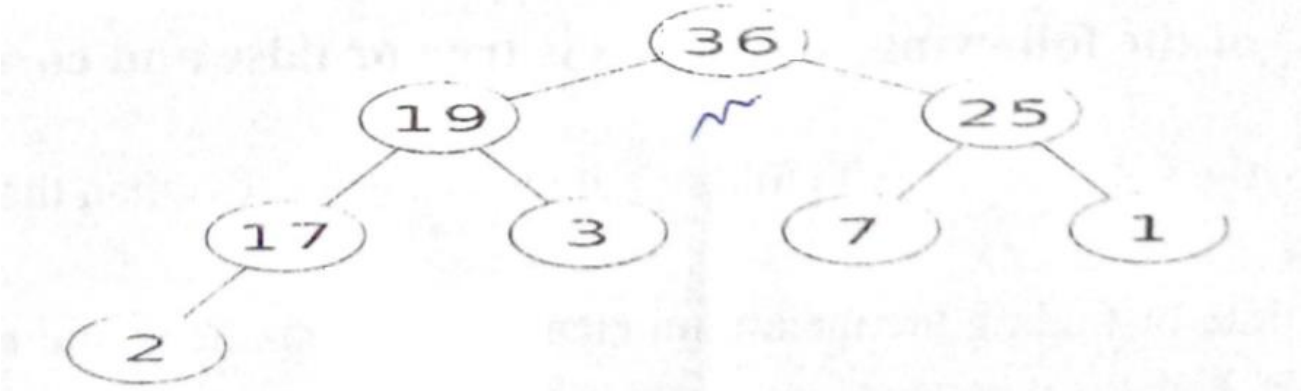
a) 100 19 36 17 3 25 1 2 7

b) 3*2+2 = 8 (index) --> 7

c)



d)



e) O(1) to find max and O(log n) to rearrange the heap

Question (5)

a) Adjacency list

Bob : Allice,3

Alice : Bob,3 --> Claire,3

Claire : Allice,3 --> Eshire,5 -->Dennis,2 -->Frank,4

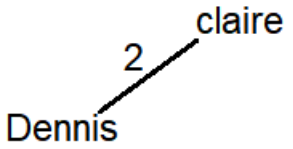
Frank : Claire,4 -->George,3

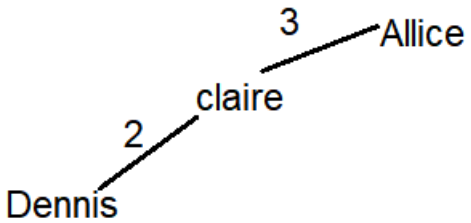
Eshire : Claire,5 -->Dennis,4

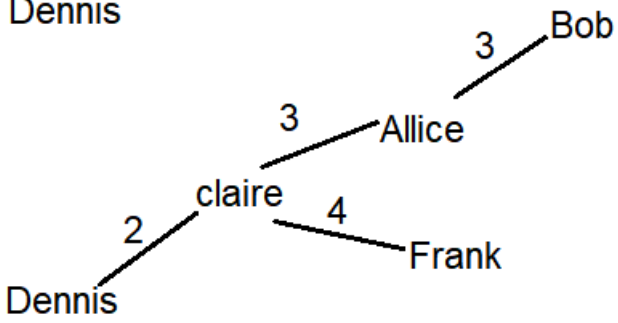
Dennis : Eshire,4 -->Claire,2 -->George,6

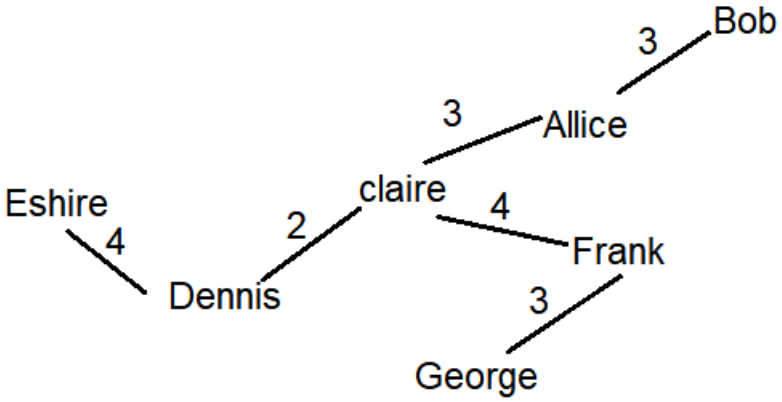
George : Dennis,6 -->Frank,3

b)

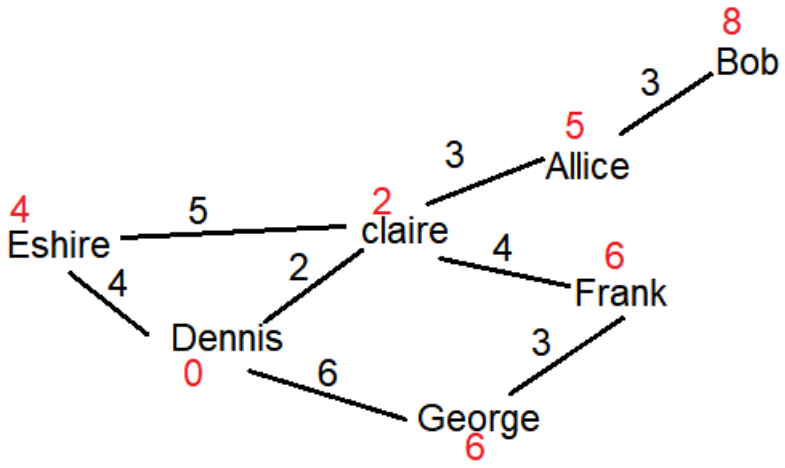








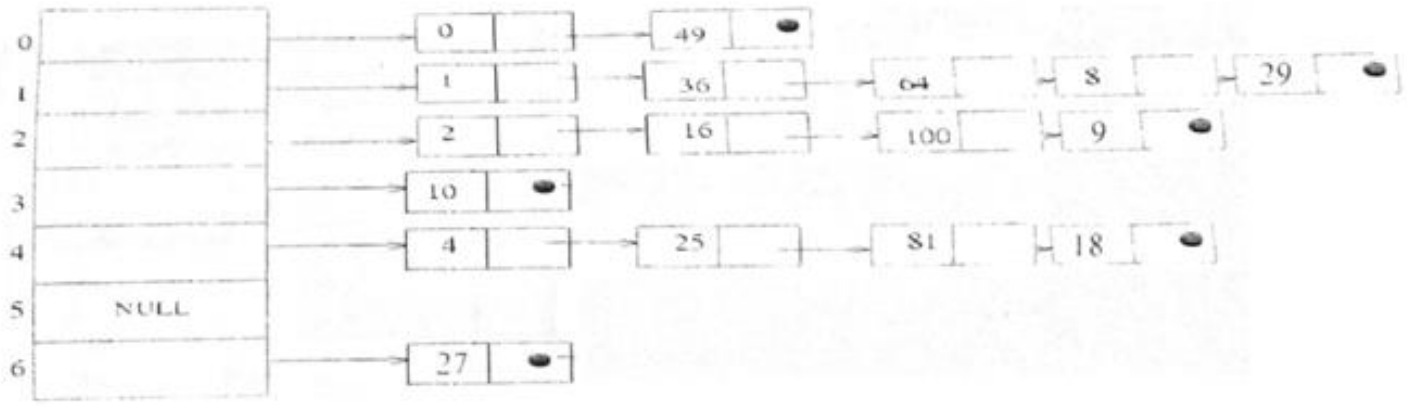
c) 5


8
Bob
3
5
Allice
3
4
Eshire
5
2
claire
4
6
Frank
4
2
Dennis
3
0
6
George
6

Question (6)

a) (2+3+2+3+4+2+3+4+2+3+4)/11 = 2.91

b)



c) Let k = {0 1 2 3 4 5 6 7 8 9}

k^2 = {0 1 4 9 16 25 36 49 64 81}

k*k%7 = {0 1 4 2 2 4 1 0 1 4}

only use 4 slots "0,1,2,4" out of 7

so k%7 which uses all 7 slots is better

d) Elements in an ascending order

0 1 2 4 16 25 36 49 64 81 100

0	0
1	1
2	2
3	81
4	4
5	25
6	64
7	100
8	
9	49
10	
11	
12	
13	
14	
15	
16	16
17	36
18	
19	

collusions

36 -> 1

64 -> 2

81 -> 2

100 -> 7

collusions = 12