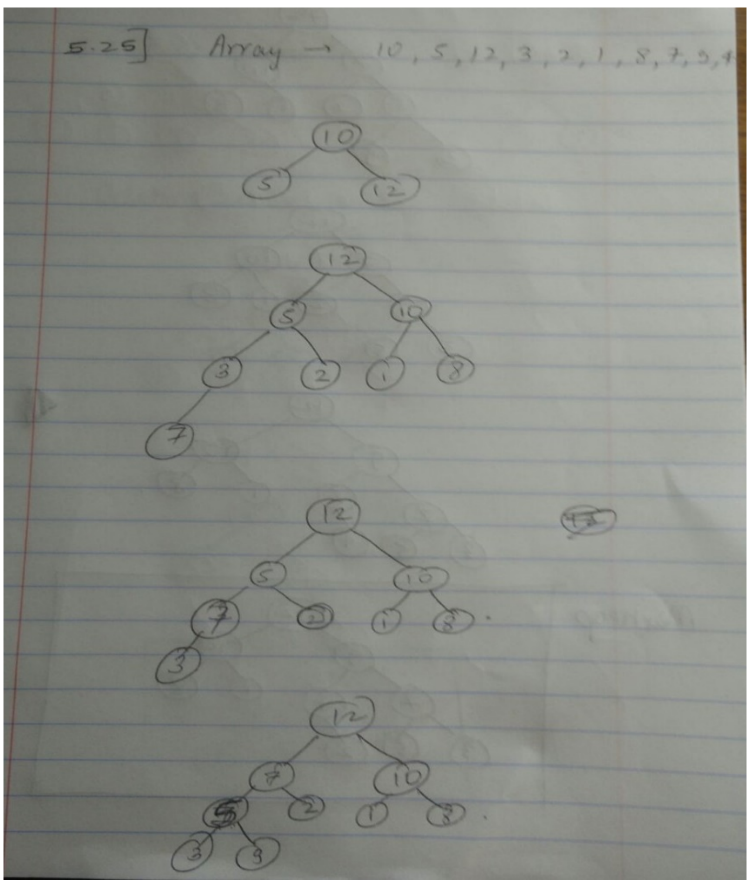
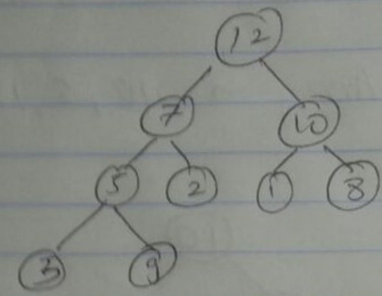
Homework 4
Mohit Galvankar mgalvank

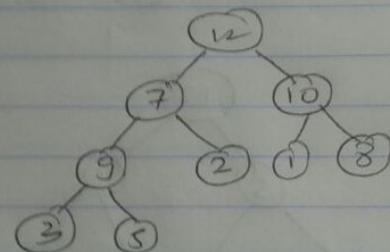
5.23 What are the minimum and maximum number of elements in a heap of height h?

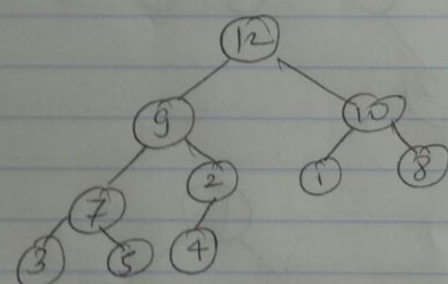
Since a heap is an almost-complete binary tree (complete at all levels except possibly the lowest), it has at most 1+2+2^2 +2^3 +…+2^h =2^(h+1)-1 elements (if it is complete) and at least 2^(h-1)+1=2^h elements (if the lowest level has just 1 element and the other levels are complete).

5.25 Show the max-heap that results from running buildHeap on the following values stored in an array:
10 5 12 3 2 1 8 7 9 4

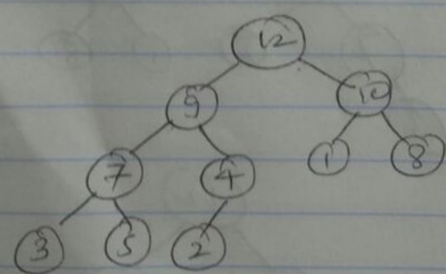




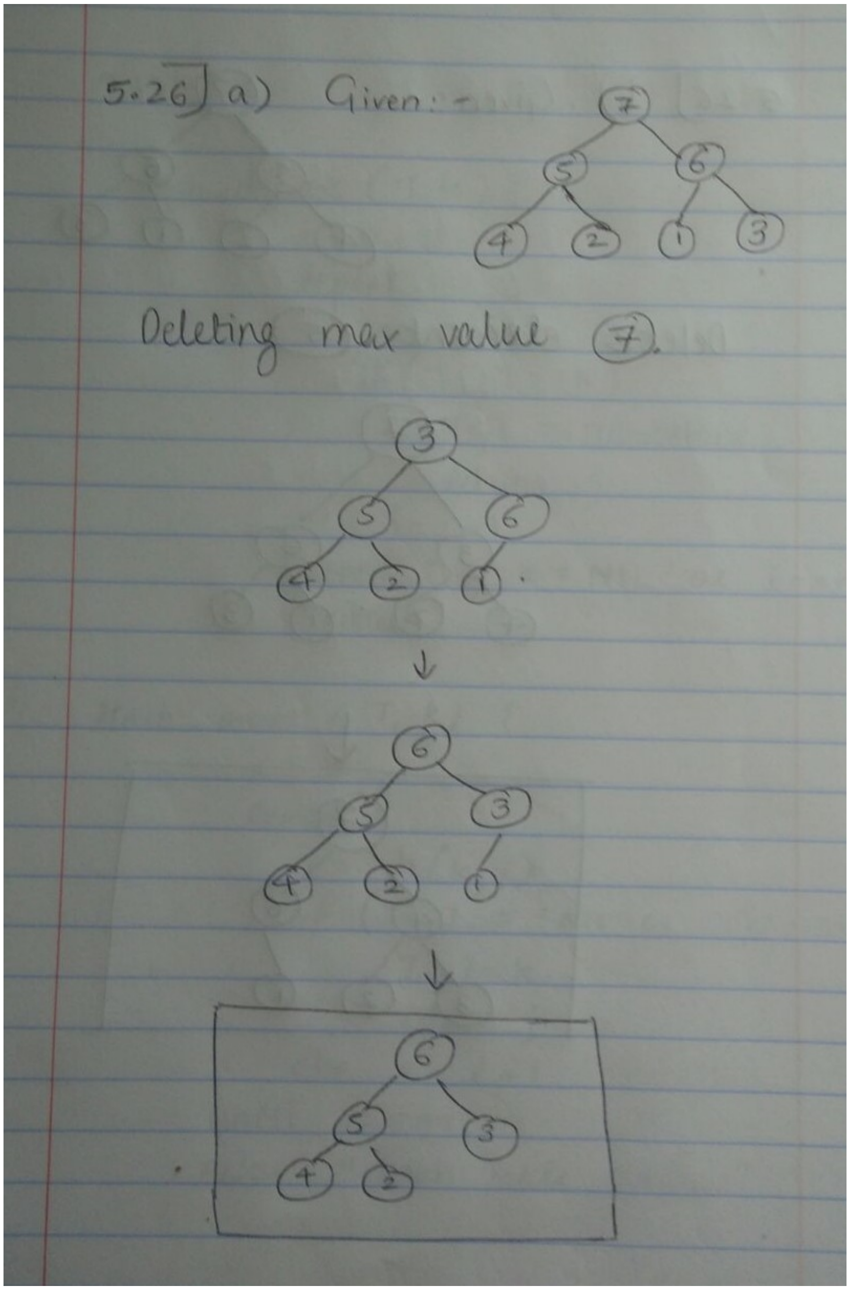




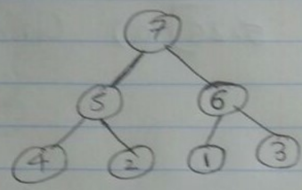
MaxHeap →



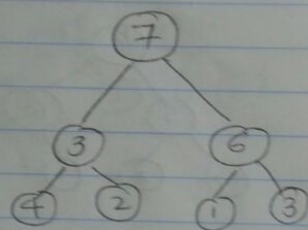
5.26 (a) Show the heap that results from deleting the maximum value from the max-heap of Figure 5.20b.
(b) Show the heap that results from deleting the element with value 5 from the max-heap of Figure 5.20b.

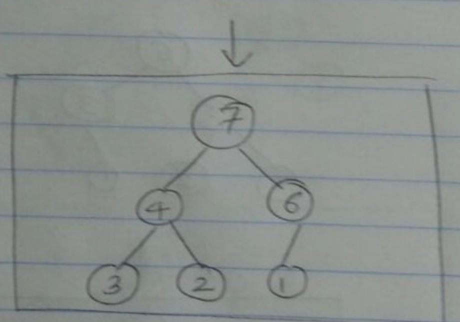


5.26] b). Given:-



Delete element 5 .





5.28 Build the Huffman coding tree and determine the codes for the following set of letters and weights:
Letter A B C D E F G H I J K L
Frequency 2 3 5 7 11 13 17 19 23 31 37 41
What is the expected length in bits of a message containing n characters for this frequency distribution?

L 00

H 010

I 011

E 1000

F 1001

J 101

D 11000

A 1100100

B 1100101

C 110011

G 1101

K 111

The average code length of a word of length n is 3.23445n.

9.13 Assume that you are hashing key K to a hash table of n slots (indexed from 0 to n □ 1). For each of the following functions h(K), is the function acceptable as a hash function (i.e., would the hash program work correctly for both insertions and searches), and if so, is it a good hash function? Function Random(n) returns a random integer between 0 and n □ 1, inclusive.

(a) h(k) = k=n where k and n are integers. No : hash value will be greater than largest index in HT when k > n^2.

(b) h(k) = 1. Yes : It would work correctly, but it is a very bad hash function because any two
items will have a collision.

(c) h(k) = (k + Random(n)) mod n. No : A hash function must return the same value each time.

(d) h(k) = k mod n where n is a prime number. Yes : In fact, this is often used in practice if not much is known about the distribution of key values.

9.16 Assume that you have a ten-slot closed hash table (the slots are numbered 0 through 9). Show the final hash table that would result if you used the hash function h(k) = k mod 10 and pseudo-random probing on this list of numbers: 3, 12, 9, 2, 79, 44. The permutation of offsets to be used by the pseudo-random probing will be: 5, 9, 2, 1, 4, 8, 6, 3, 7. After inserting the record with key value 44, list for each empty slot the probability that it will be the next one filled.

9.16] Given :- 3, 12, 9, 2, 79, 44.

pseudo-random probing list → 5, 9, 2, 1, 4, 8, 6, 3, 7

Hash function h(k) = k mod 10.

0	
1	
2	12
3	3
4	44
5	79
6	
7	2
8	
9	9

Steps

3 mod 10 → 3 insert at index 3.

12 mod 10 → 2 insert at index 2.

9 mod 10 → 9 insert at index 9.

2 mod 10 → 2 collision ~~2mod10+5~~ (2 + 5) mod 10

79 mod 10 → 9 collision (79 + 4) mod 10 → 3

collision

(79 + 6) mod 10 → 5

insert at index 5.

44 mod 10 → 4 insert at index 4.

9.19 Write an algorithm for a deletion function for hash tables that replaces the record with a special value indicating a tombstone. Modify the functions hashInsert and hashSearch to work correctly with tombstones.

9.19]

```
Hash_delete (T, k) {
        i = 0
        repeat
            j = h(k, i)
            if (T[j] == k)
                T[j] = Tombstone
                return
            i = i + 1
        until T[j] == NIL or i == m
        return

Hash_insert ( T, k) {
        i = 0
        repeat
            j = h(k, i)
            if (T[j] == NIL or T[j] == Tombstone)
                T[j] = k
                return j
            else i = i + 1
        until i == m
        error "hash table overflow"
```

```
Hash_search (T,k) {
    i=0
    repeat
        j = h(k,i)
        if ( T[j] == k or  T[j] == tombstone )
            return j
        else  i = i+1
    until i==m
    error " Not found".
```

13.1 Show the binary trie (as illustrated by Figure 13.1) for the following collection of values: 42, 12, 100, 10, 50, 31, 7, 11, 99.

13.1] Given:- 42, 12, 100, 10, 50, 31, 7,
11, 99.