2) (10 pts) ALG (Heaps)

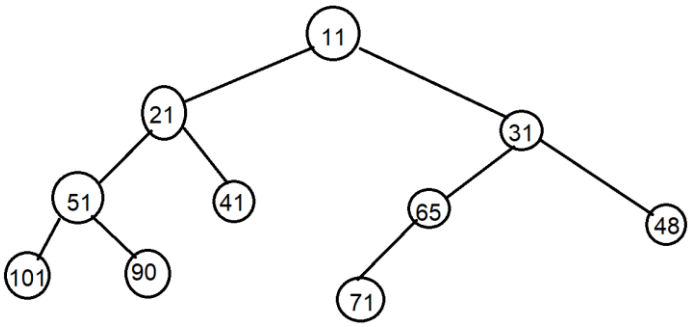
(a) (3 pts) A heap is represented by the array below. The first item is stored at index 1. Answer the following questions (**please answer the data not the index where it's stored.)**

index	1	2	3	4	5	6	7	8	9	10
data	7	11	13	16	18	19	24	21	20	35

i). Who is the left child of 13: **_19**____, ii). Right child of 16: **_20**__iii) parent of 24:__**13**____

Gade: 3 pts (1 pt for each)

(b) (2 pts) Consider the following tree. Is this a valid minheap? Justify your answer. *Just saying yes/no has no credit without justification.*



It is not a valid minheap because it's not a complete binary tree. Specifically, 41 has no left child but 65 has a child, so the nodes on the bottom most level aren't filled out left to right.

Grade: 2 pts all or nothing.

(c) (5 pts) Consider a **minheap** stored in an integer array *int heaparray[100]*, which is globally declared. Write a percolateUp function that takes an index and perform the full percolate up operation for the item at that index. While writing the code, you can assume that there is a swap function available for you that is described below.

```
// swap(int* ptrA, int* ptrB) - swaps the contents in the variables
//                              pointed to by ptrA and ptrB.
void percolateUp(int idx){

    if (idx > 1) {   // 1 pt
        if (heaparray[idx/2] > heaparray[idx]) { // 1  pt
            swap(&heaparray[idx], &heaparray[idx/2]); // 2 pts
            percolateUp(idx/2);   // 1 pt
        }
    }

}
```