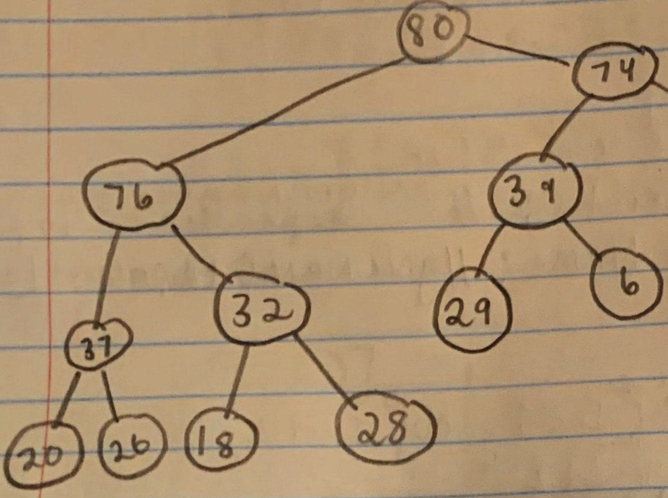
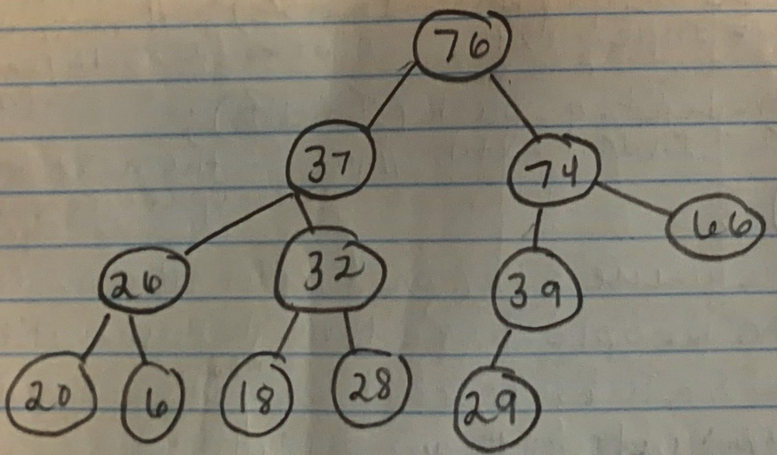
Lauren Magee
Ben Teig

CS 303 HW #6 (odds)

① Show the result of removing the #80 from the following heap. Show the new heap and its array representation.





Array: 76 ,37, 74, 26, 32, 39, 66, 20, 6, 18, 28, 29

③ Write a compare function class that inserts Person objects in a priority queue based on the number of dependents a person has. The person object with the largest # of dependents should be removed first.

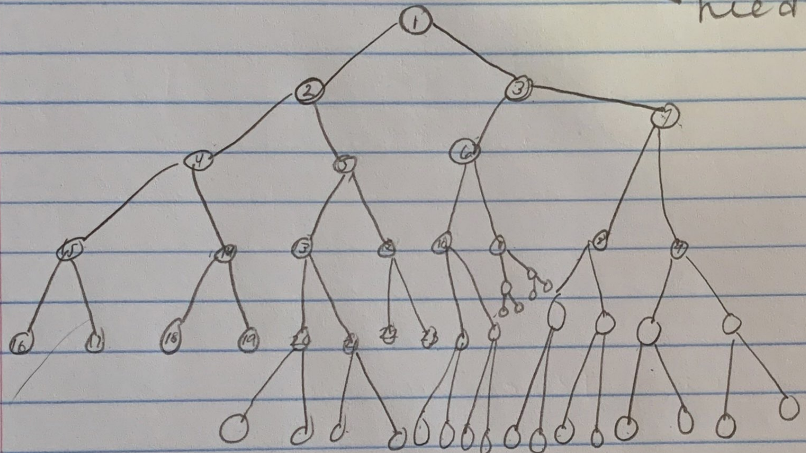
```
class Person {
    public:
        int dependents; // assign number of dependents to person
        string personName; //give Name to parent|guardian
};

 int compare (obj Ind1, obj Ind2) {
    if ( Ind1 < Ind2) {
        dependent_q.push( Ind 2); // so the most dependents is removed first
        Ind 2 = Ind2.next();
    }
   else{ dependent_q.push(Ind1);
        Ind 1 = Ind1.next(); //gives it the next object to compare to the remaining object
}
int main() {
    priority_queue < person, vector <Person>, Compare>
    // create people (objects) here through Person class    dependent_q;
    // assign values to each object's dependents
    int individuals = # of objects;
    while (priority_queue.size() != individuals) {
        compare( ind 1, ind 2); // Loops until all objects are inserted into the queue
    }
```

⑤ What would the Huffman code look like if all symbols in the alphabet had equal frequency?

alphabet = 26 letters so <u>26 child nodes</u>

need for 26 different results



∴ Turns out to be a complete binary tree