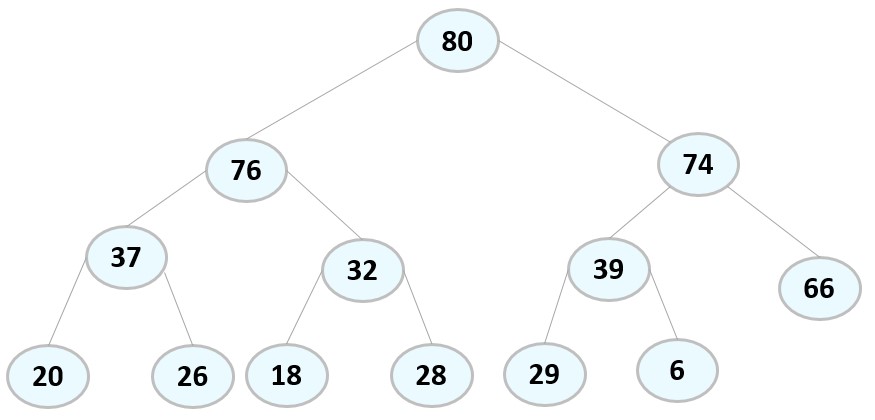
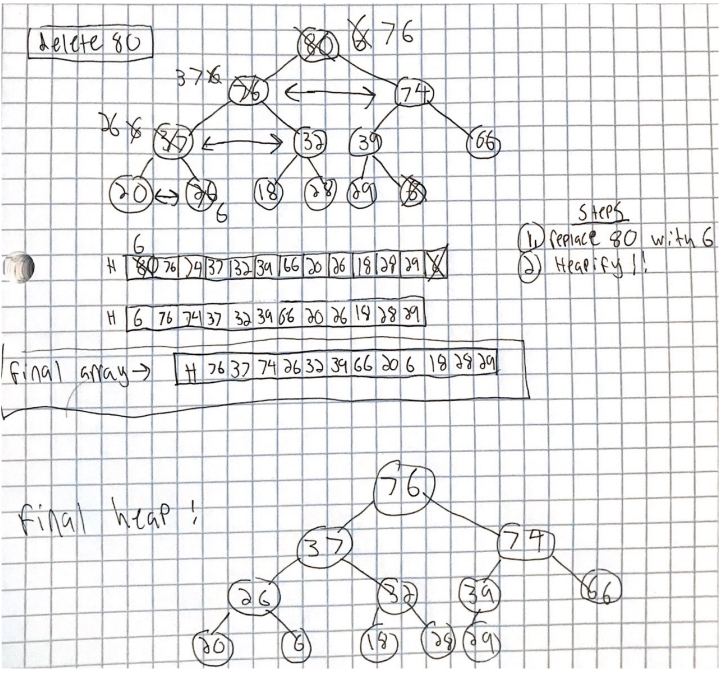
Assignment 5

1. (Weight: 20%) Show the result of removing the number 80 from the following heap. Show the new heap and its array representation.





1. (Weight: 10%) The largest object should be removed first from a priority queue. Explain why operator less is used as the default to compare objects being placed in a priority queue instead of operator greater.

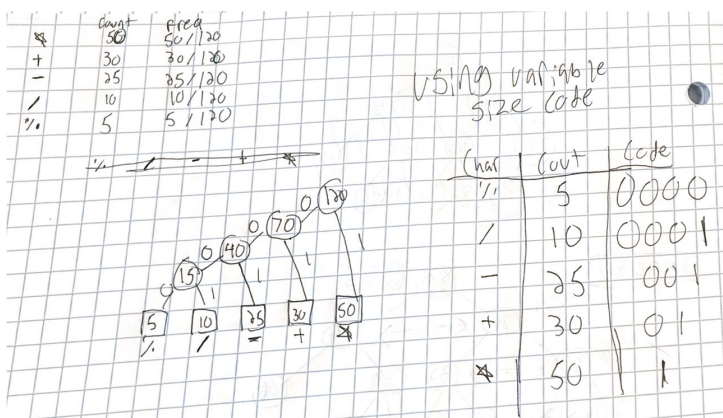
We use the “less” operator because that is the condition that we use to determine if an item is to be swapped. If we used the “greater” operator, the items would only swap if the parent was less than the child. This would turn the max-heap into a min-heap and it would store the item of the lowest priority at the front of the queue.

1. (Weight: 20%) Programming: 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 number of dependents should be removed first.

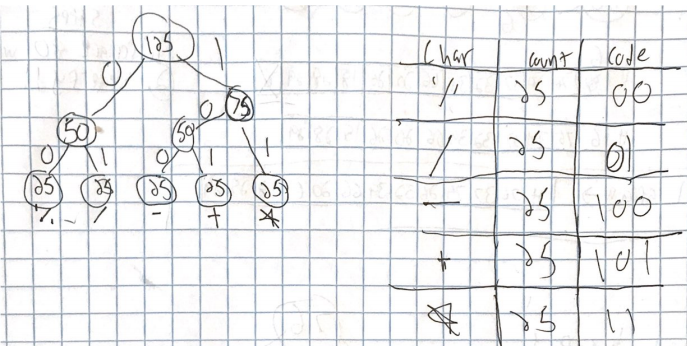
|  |
| --- |
| class Compare\_Priority\_Queue {  bool operator()(const Binary\_Tree<Person> & left\_tree, const Binary\_Tree<Person> & right\_tree) {  int wLeft = left\_tree.get\_data().dependents;  int wRight = right\_tree.get\_data().dependents;    return wLeft < wRight;  }  }; |

1. (Weight: 20%) Create the Huffman code tree for the following frequency table:

|  |  |
| --- | --- |
| Symbol | Frequency |
| \* | 50 |
| + | 30 |
| - | 25 |
| / | 10 |
| % | 5 |

stream

1. (Weight: 10%) What would the Huffman code look like if all symbols in the alphabet had equal frequency?



1. (Weight: 20%) Programming: Use the generic Huffman\_Tree<T> on Blackboard to encode the words in your favorite nursery rhyme.