Module 2 Problem Set

1. [20 pts] Suppose that we choose 3 as our factor in the "Big Box Store" algorithm. That is, when the stack is full we allocate 3x the amount of current storage. What is the complexity of M operations in this case? [Big Oh answer].

2. [20 pts] Repeat question 2 with a factor of 11. That is, when the stack is full, allocate 11x the current size.
What is the complexity of M operations in case where factor = 11?
Give your answer (and the associated reasoning) in Big Oh notation.

3. [20 pts] For this problem, let i be the sets in a union/find problem.
Let f[i] be the find array associated with this union/find problem.

Draw the tree associated with the f array below:

```
          i     1 2 3 4 5 6 7 8 9 10
                ----------------------
       f[i]     2 2 4 2 6 7 2 4 5 6
```

4. [20 pts] Can the find array of problem 5 be a result of running weighted quick union?
Explain why this is impossible OR ELSE give a sequence of union-find operations that produce the above table.

5. [20 pts] Suppose we introduce the "add_new_set()" operation. The operation will return a new whole number corresponding to a new disjoint set. This means that the number of nodes in our graph may change dynamically. Assuming add_new_set is O(1), what is the complexity of a sequence of add_new_set, union, and find operations?

Note: You should use weighted quick union and find with path compression to implement the union and find operations