1. Big-Oh
   a. Complexity of a find() in a union-find data structure containing N elements (no path compression). (worst case) 
      \( O(N) \)
   b. Complexity of push() onto a stack containing N elements implemented with a linked list. (worst case) 
      \( O(1) \)
   c. \( f(n) = N \log^2 N + N^2 \log N \)
      \( O(N^2 \log N) \)
   d. Complexity of a preorder traversal of a Binary Search Tree containing N elements (worst case) 
      \( O(N) \)
   e. Complexity of an IncreaseKey(k, v) on a binary min heap containing N elements. Assume you have a reference to the key k. v is the amount that k should be increased. (worst case) 
      \( O(\log N) \)
   f. \( f(n) = N! + 2^n \)
      \( O(N!) \)
   g. int example (int n) {
       int sum = 0;
       for (int i = 0; i < n*n; i++) {
           for (int j = i; j > 0; j=j/2) {
               sum++;
           }
       }
       return sum;
   }
      \( O(N^2 \log N) \)

2. Proving Big-Oh (c, and n0)
   Suppose \( f(n) = 12n^2 + 42n - 3 \). Prove that \( f(n) \) is \( O(n^2) \) using the definition \( f(n) \) is \( O(g(n)) \) if there exists constants \( c \) and \( n0 \) such that \( f(n) \leq g(n) \) for every \( n \geq n0 \).
   Let \( n_0 = 1, \) \( 12n^2 + 42n - 3 \leq c n^2 \)
   \( 12(1) + 42(1) - 3 \leq c(1) \)
   \( 51 \leq c \)
3. **Best Data Structure**
For each of the following tasks, what is the most efficient structure and the worst-case time complexity for performing the operations on the structure. You may choose from the following: sorted array, sorted linked list, binary search tree, AVL tree, min heap, up tree, stack implemented with a linked list, queue implemented with an array.

a. Keeping track of next customer at a sandwich shop, with new orders constantly coming in.
   - Queue, $O(N)$

b. Finding the next patient to be examined in an emergency room.
   - Min Heap, $O(1)$

c. Finding and removing the minimum value.
   - Sorted Linked List, $O(1)$

d. Inserting a new value.
   - Stack, $O(1)$

4. **AVL insertion**
Insert the following values into an AVL tree (starting with an empty tree). 7, 2, 3, 8, 16, 25. **Circle your final answer if intermediate steps are shown.**
5. Number of Nodes in Various Types of Trees
   a. What is the minimum and maximum number of nodes in a **binary search tree of height 6**? (Hint: the height of a tree consisting of a single node is 0) Give an exact number.

   Minimum = 7
   Maximum = 127

   b. What is the minimum and maximum number of nodes in a **complete binary tree of height 5**?

   Minimum = 32
   Maximum = 63

6. Proof

Prove by induction that the sum of numbers from 0 to N is equal to \( N(N + 1)/2 \)

\[
0 + 1 + 2 + 3 + 4 + \ldots + N = \frac{N(N+1)}{2}
\]

**Base Case:** \( N = 0 \)

\[
0 = (0)(0 + 1)/2
\]

0 = 0

**Induction Hypothesis:** \( N = k \)

Assume the formula to be true for all \( k, 0 \leq k \leq N \)

**Induction Step:** \( N = (k + 1) \)

\[
\text{Sum}(0 \text{ to } k+1) = 0 + 1 + 2 + 3 + \ldots + k + (k+1)
\]

\[
= \text{Sum}(0 \text{ to } k) + (k+1)
\]

\[
= k(k+1)/2 + (k+1) \quad \text{[Induction Hypothesis]}
\]

\[
= k(k+1)/2 + 2(k+1)/2
\]

\[
=(k+1)*(k+2)/2
\]

\[
=(k+1)*((k+1)+1)/2
\]

Proven.