**6.14 Possible Subsets in Lexicographical Order**

**Aim:** The aim is to write an efficient program that generates all possible subsets (the power set) of a given set in lexicographical order.

**Algorithm:** 1. Let the input be a set of elements:  
  S = {s₁, s₂, …, sn}

2. Sort the elements of S in non-decreasing order.  
  (This ensures that subsets are generated in lexicographical order.)

3. The total number of subsets of S is 2ⁿ.

4. We define a recursive backtracking procedure:  
  GenerateSubsets (start, current, result)  
  - start → the index in S to consider next  
  - current → the subset being built  
  - result → collection of all generated subsets

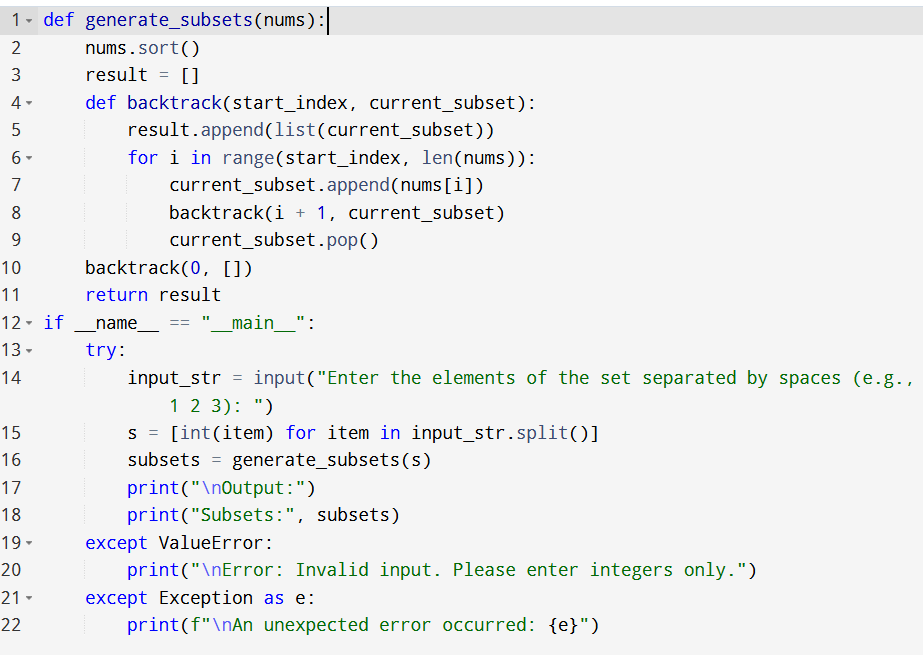
5. At each call:  
  - Add a copy of current to result (every intermediate state forms a valid subset).

6. Recursive case:  
  - For each index i from start to n−1:  
    a. Include S[i] in current.  
    b. Recursively call:  
      GenerateSubsets (i+1, current, result)  
    c. Backtrack by removing S[i] from current.

7. Initially, call the procedure as:  
  GenerateSubsets (0, ∅, result)

8. After recursion completes, result will contain all subsets of S.  
  Because the input was sorted and recursion progresses left-to-right, subsets appear in **lexicographical order**.

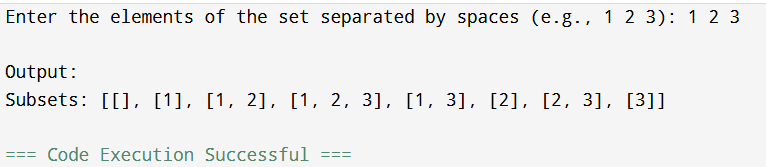
**Program:**

****

**Input:**

* Enter the elements of the set separated by spaces (e.g., 1 2 3):

**Output:**

****

**Result:** Thus, the program is executed successfully and output is verified.

**Performance analysis:**

* Time Complexity: O(n\*2^n)
* Space Complexity: O(n).