ADA Assignment 01

Calculating Time Complexities

Chandrajeet Nagar

B13115
CSE

Objective

Study the running times of each program.

1. Write a program to accept an integer n > 0, and generate all permutations between 1 to n.
2. Write a program to accept a set of n elements, and generate all the subsets of that set. Assume that all elements of the set are characters.

Language and Predefined Functions used

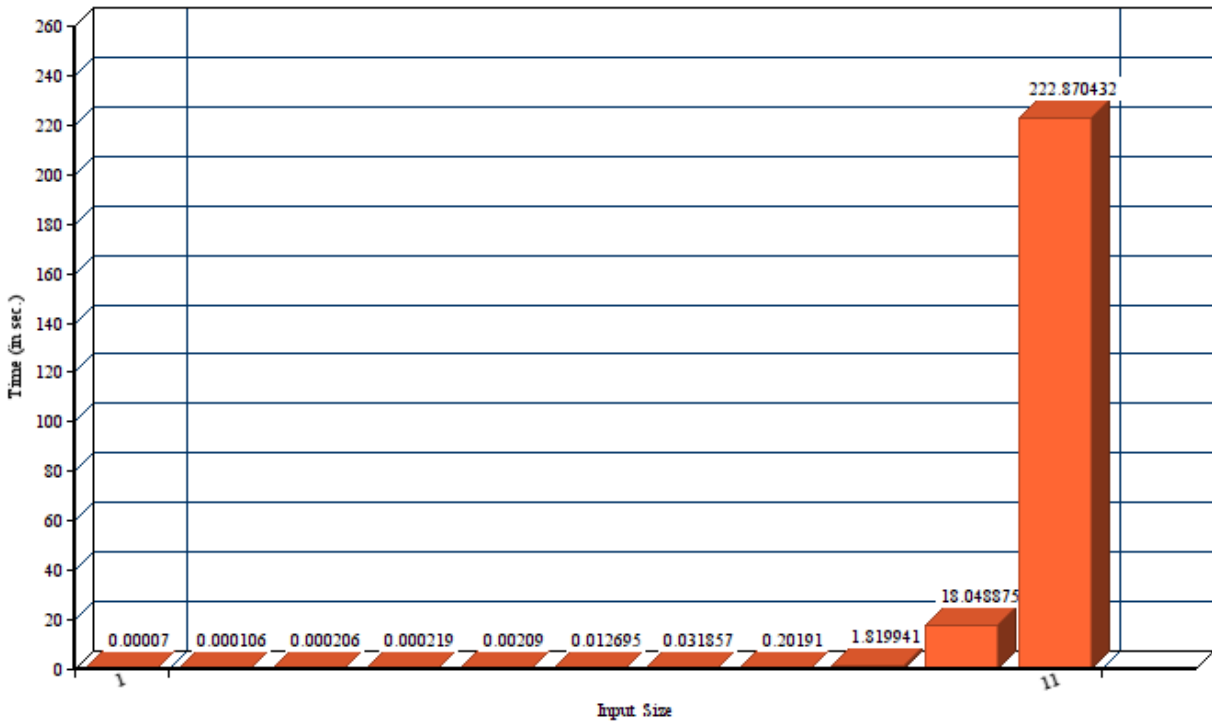
I have used python.

Predefined Functions:

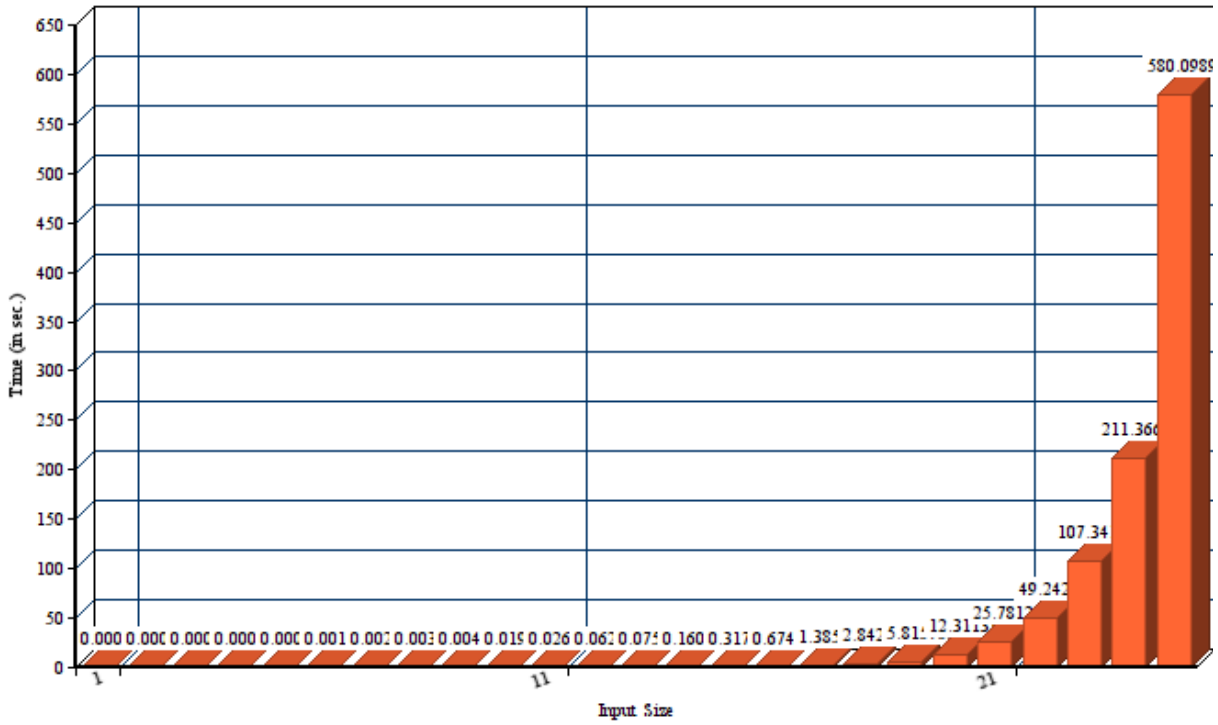
1. pow(a,b) : Calculates a^b. It uses near constant time complexity[1] i.e. O(1).
2. len(arr) : It returns the length of the array. It uses linear time complexity i.e. O(n).

Methodology

1. There are n! possible permutations of n. I have made an array of size n and generated its permutation by swapping its starting index and working index then recursively calculating permutation of rest of array and checked if starting index is equal to ending index then printing the generated array and then backtracked it to original array recursively. I have taken value of n from 1 to 12 as it was taking too much time for n> 12. Since I am generating every permutation and printing it so the Asymptotic complexity is O(nn!).



2. There are 2^n possible subset of given set. First I have append 0 in array of size 2^n and adding 1 each time wherever i got 1 after adding in array i have just compare to string and print respective element of string and for all value of array was equal to zero i have simply put “Null” there.I have taken length of string ranging from 1 to 25 as it was taking too much time for length >25.In part B the Asymptotic complexity is of order O(n2^n).



CONCLUSION

After calculating b0th question' s time complexity i have got both graph exponentially increasing.I have noticed graph of nn! more increasing monotonically then n2^n graph for same input size.So,i have noticed that for an algorithm O(n!) time complexity worse then O(2^n).