Homework

Nguyen Pham Phuong Nam - 23520978
Pham Huynh Long Vu - 23521813

7 November 2024

Problem 1: Given an input string of numbers, find all combinations of numbers that can be formed using digits in the same order.

Complexity Analysis:

- **Number of combinations:** For n digits, there are $2^n - 1$ non-empty subsets.
- **Time complexity:** $O(n \cdot 2^n)$ because generating each subset requires $O(n)$ time.
- **Space complexity:** $O(n)$ for storing temporary strings.

Solution:

Iterate through all subsets using binary representation (bitmask). Ensure the order of digits is preserved.

Problem 2: Given a set of characters and a positive integer k, print all possible strings of length k that can be formed from the given set.

Complexity Analysis:

- **Number of strings:** For a set of size m, there are m^k strings of length k.
- **Time complexity:** $O(m^k)$ since generating each string requires $O(k)$ time.
- **Space complexity:** $O(k)$ for storing temporary strings.

Solution:

Use recursion; at each step, iterate through all characters and append to the current string until the length reaches k.

Problem 3: Write a program to print all the combinations of factors of given number n.

Complexity Analysis:

- **Number of combinations:** Approximately $O(2^{\sqrt{n}})$ as n has at most $\sqrt{n}$ divisors.
- **Time complexity:** $O(2^{\sqrt{n}} \cdot \sqrt{n})$ since each combination requires $O(\sqrt{n})$ time to verify.
- **Space complexity:** $O(\sqrt{n})$ for storing the divisors.

Solution:

Use recursion to iterate through all combinations of divisors of n in ascending order.

Problem 4: Count the number of ways to represent x as the sum of the nth powers of natural numbers

Complexity Analysis:

- **Number of combinations:** Depends on $x^{1/n}$, the maximum number of natural numbers that can be used.
- **Time complexity:** $O(x^{1/n})$ because each number can participate in a combination.
- **Space complexity:** $O(n)$ for storing the recursion state.

Solution:

Use recursion to check all combinations with the constraint of not reusing the same number.

Problem 5: Tower of Hanoi

Complexity Analysis:

- **Number of moves:** For n disks, the minimum number of moves is $2^n - 1$.
- **Time complexity:** $O(2^n)$ since each disk requires two moves (to the intermediate and destination rods).
- **Space complexity:** $O(n)$ for storing the recursion state.

Solution:

Use recursion:

- Move $n - 1$ disks from the source rod to the intermediate rod.
- Move the largest disk from the source rod to the destination rod.
- Move $n - 1$ disks from the intermediate rod to the destination rod.

.