Problems based on Recursion - 9

Assignment Solutions

PW SKILLS

Q1 - You are given a string. Your task is to divide the string into palindromic substrings. Print all such partitions.

Example

Input: banana

Output

[b, a, n, a, n, a]
[b, a, n, ana]
[b, a, nan, a]
[b, ana, n, a]
[b, anana]

Explanation

- Create a recursive function that has 4 parameters-
  - s - the input string.
  - ans - the current set of palindromic subsets that end before the index start.
  - start - the index at which current palindromic substring starts.
- Base case - start reaches the end of the string. In this case we print the current set of palindromic substrings i.e. ans.
- At each call we try to generate a substring that starts at the index 'start' that is a palindrome. If we find such a substring, we append it to our ans, and call the function for the remaining string.
  - To check if the current substring is a palindrome or not, we can use a two pointer approach where one pointer will start from the left end of the string and the other will start at the right end, and slowly they will move towards each other till they pass the other. If at any point the characters pointed to by the pointers don't match then the string is not a palindrome, otherwise it is.

https://pastebin.com/0cz8SVVC

```
C:\Users\akshi\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\I
banana
[b, a, n, a, n, a]
[b, a, n, ana]
[b, a, nan, a]
[b, ana, n, a]
[b, anana]

Process finished with exit code 0
```

Q2 - A string is called beautiful if is an even length string consisting of only stars('*') and dashes('-'). Further the number of stars in the first half of the string should be equal to the number of stars in the second half of the string. Given a number n, print all the beautiful strings of length 2 * n.

Example

Input: 2

Output

```
----
*-*-
*--*
-**-
-*-*
****
```

Explanation

The approach is divided into 2 steps-

1. Generate all the beautiful strings of length n and divide them into groups according to the number of stars that they contain.
   a. Create a recursive function that has 4 parameters-
      i. remainingChars - number of characters to be inserted to make our current string of length n.
      ii. nums - 2d matrix. Each row contains strings that have the strings with the number of stars equal to the row number. Ex- row 0 has strings with 0 stars, row 1 has strings with 1 star etc.
      iii. numberOfStars - the number of stars in the current string.
      iv. curr - string formed so far.
   b. Base case - remaining characters become 0. In this case add the current string into the nums matrix depending on the number of stars it has.
   c. Since we are generating all the patterns possible, at each call we can add either a star or a dash to our string and call the recursive function on it.
2. Make pairs of strings that have an equal number of stars present in them. Concatenate them and print.
   a. For each row of the nums matrix, call the print function.
   b. The print function uses 2 nested for-loops to concatenate each string in the arraylist, with every other string in the same arraylist, and print them.

https://pastebin.com/q6N1Rzxu

```
C:\Users\akshi\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains
2
----
*-*-
*--*
-**-
-*-*
****

Process finished with exit code 0
```

Q3 - A string is called beautiful if it consists of only stars('*') and dashes('-'). Further the number of stars in the first half of the string should be equal to the number of stars in the second half of the string. Given a number n, print all the beautiful strings of length n. If the value of n is odd, the middle value can be either '*' or '-'

Example

Input: 3

Output

```
-*-
---
***
*-*
```

Explanation

The approach of this question is same as previous one except-

1. The recursive function will create strings of length n/2.
2. The print function for odd length will add a dash and a star in the answer string.

https://pastebin.com/0y7Dw0Er

```
C:\Users\akshi\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrai
3
-*-
---
***
*-*

Process finished with exit code 0
```

Q4 - Problem Count the number of substrings having same first and last characters

Input : s = "pqrpq"
Output : 7

Explanation:

- To find whether the substrings have the same first and last characters, we have to find all the substrings first using recursive function count().
- Then, we'll have to compare the first and last elements of every substring and if equal then increment the variable ans and in the end return ans.
- There are 2 base cases: if length of string is 1 then increment the count as a string of length 1 has the same first and last character., if string is empty then return 0.
- Else call the recursive function count.

Code

https://pastebin.com/WS5f6nkj

```
C:\Users\akshi\.jdks\openjdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\Ir
pqrpq
7

Process finished with exit code 0
```

Q5 - You are given a string s. All the characters in s are distinct. Your task is to generate all the strings that contain the characters of 's' in increasing order.

Example

Input

sam

Output

a
am
ams
as
m
ms
s

Explanation

1. Sort the characters of the string so that we get them in increasing order.
2. Generate all the substrings-
   a. Create a recursive function that takes in 4 parameters-
      i. str - sorted string
      ii. n - size of the original string
      iii. index - current index
      iv. curr - substring formed so far
   b. Base case- current index is equal to n. In this case simply return back.
   c. In any call, first print the substring formed so far. Then traverse the remaining characters of the sorted string.
      i. At each iteration add the current character to the current substring and call the recursive function with the updated index and current substring.
      ii. After the call, remove the recently added character from the current substring to backtrack to the initial state.

Code

https://pastebin.com/jhFk7SdE

```
som

a
am
ams
as
m
ms
s

Process finished with exit code 0
```