

Problems based on Recursion - 6

Assignment Solutions



Q1 – Given an array arr of size N and a target value target The task is to find all the indices of the given target value in the array.

(Medium)

Input: arr = {1, 2, 9, 2, 2, 9}, target = 2

Output: 1 3 4

Element 2 is present at indices 1, 3, 4 (0 based indexing)

Input: arr[] = {8, 8, 8}, target = 8

Output: 0 1 2

- We have created a “allIndex” function which will return all indices of the given target value in the array so its return type is array.
- In this function we have 3 parameters, the array, a target value and the current index.
- The return type for this function is “array of int type”, it returns the array containing all indices of the given target value, if the target is not available in the given array it will return an empty array.
- We do not know currently whether we have the target value existing in the array or not. All we know is the current value.
- The base case or terminating condition is that if the current index ‘current’ overshoots the length of the array that means we have no further element to check on for the target existence. So, If the current index reaches the length of the array, then return empty array.
- Else check the answer for the vector from “current index +1” to “last index” and store the answer in the vector “smallIndex”.
 - If the element at the current index is not equal to target then just simply return the answer which came from recursion.
 - Else if the element at current index element is equal to target then create a new vector of size 1 greater than the “smallIndex” vector (which came from recursion) and store the current index as its first element and rest of the vector as same as “smallIndex” vector(which came from recursion)

Code: <https://pastebin.com/qV9TwRCw>

Output:

```
1 3 4
...
...Program finished with exit code 0
Press ENTER to exit console.[]
```

Q2 – Given an array of integers, print a sum triangle using recursion from it such that the first level has all array elements. After that, at each level the number of elements is one less than the previous level and elements at the level will be the Sum of consecutive two elements in the previous level.

(Medium)

Sample Input: [5,4,3,2,1]

Sample Output:

```
5, 4, 3, 2, 1  
9, 7, 5, 3  
16, 12, 8  
28, 20  
48
```

Code:

<https://pastebin.com/r3ajZp2k>

```
Enter the size : 5  
Enter the array elements : 5 4 3 2 1  
5, 4, 3, 2, 1  
9, 7, 5, 3  
16, 12, 8  
28, 20  
48
```

Q3 – Given an array of size n, generate and print all possible combinations of r elements in array.

(Hard)

Input:

```
n = 4  
{1, 2, 3, 4}  
r = 2
```

Expected Output:

```
{1, 2}  
{1, 3}  
{1, 4}  
{2, 3}  
{2, 4}  
{3, 4}
```

Explanation:

- We create 2 functions.
- printCombination creates a temporary array of length r to store all combination one by one and then calls the recursive function combination.
- The recursive function has parameters as input array, n, r, index for temporary array, temporary array, i for input array.
- If index == r, it means the current combination is ready to be printed, so we print it.
- If i is greater than or equal to n, then no more elements are there to put in temporary array, return.

- Next, We one by one consider every element of input array, and recur for two cases:
 - i. The element is included in current combination (We put the element in data[] and increment next available index in data[]])
 - ii. The element is excluded in current combination (We do not put the element and do not change index)
- When number of elements in data[] become equal to r (size of a combination), we print it.

Code:

<https://pastebin.com/n1PyUqb3>

```
Enter the length of array : 4
Enter the elements of array : 1 2 3 4
2
1 2
1 3
1 4
2 3
2 4
3 4

...Program finished with exit code 0
Press ENTER to exit console.[]
```

Q4 – Given two sorted arrays A and B of length m and n respectively, generate all possible arrays such that the first element is taken from A then from B then from A, and so on in increasing order till the arrays are exhausted using recursion. The generated arrays should end with an element from B.

(Hard)

Input:

m = 3
n = 4
A = {10, 15, 25}
B = {1, 5, 20, 30}

Expected Output:

10 20
10 20 25 30
10 30
15 20
15 20 25 30
15 30
25 30

- We create a recursive function with parameters as both the arrays, the third array which will store all the combinations, two pointers, i and j to keep track of indices of array A and B respectively, length of all three arrays, and a flag variable.
- Flag variable is of boolean type, it indicates whether current element in output should be taken from 'A' or 'B'.
- In the function, first check whether flag is true or false.
- If flag is true, first check if length of output array is greater than 0 which indicates that we have one of the combinations, and we print it using a printArr function we create.
- Next, while flag is still true, we continue the combination in the same array by traversing over elements of first array from its current index.
- If length of output array is 0, it means we are adding the first element, so directly add current element from first array to the output array at its current index.
- Give a recursive call with change in parameters: index of first array as current index +1 and reverse flag.
- If length of output array is not 0, we need to check if element at current index of output array is less than element at current index of first array, only then we add it to output array, and give a recursive call with change in parameters: index of first array as current index +1, index of output array as current index +1 and reverse flag.
- If flag is false, we need to add element from second array.
- Traverse through the second array, if element at current index of output array is less than element at current index of second array, only then we add it to output array, and give a recursive call with change in parameters: index of second array as current index +1, index of output array as current index +1 and reverse flag.

Code:

<https://pastebin.com/urKv5ayB>

```
Enter the length of both arrays : 3 4
Enter the elements of first array : 10 15 25
Enter the elements of second array : 1 5 20 30
10 20
10 20 25 30
10 30
15 20
15 20 25 30
15 30
25 30
```

```
...Program finished with exit code 0
Press ENTER to exit console.█
```