Report: Assignment 1(MTL 342)

Submitted by: Abhishek Yadav(2015MT10585) and Barinderpreet Singh(2015MT10594)

Subject: Printing no. of permutation not possible using an IRD.

Algorithm:

Basically, we have used brute force technic, we first generated all the permutation possible for a given value of 'n'. Then we passed them in a function to check if they are possible to generate using an "IRD" or not. For that I just took three Linked list type data structure and in first one I kept the permutation to generate and the second one was to act as my IRD and third one just contained the numbers from 1 to n.

Then, when I had to output a particular number, I first matched the first number of first and third list, if the first number of the permutation which I had to produce was that is my first list was more than the third one, I just pushed every element into second list till the first number of first list was greater than the first element of the third list.

Then I just removed the first element from first and third list and returned the list again to the function that is I did the same steps recursively.

If the element was first element of the first list was less than the first element of third list that means I need to pop, so I checked if the first or last element of the second list was same as the first element to be popped now and I removed that if I found and returned the list again, if that was not the case then I returned false, as the permutation was not possible to produce.

If the size of first list was zero I returned true;

Complexity: Basically I checked for all the permutation, so it took about n! Times for every possibility and also it took about O(n) time to check for if it is possible or not.

So total complexity was O(n!*n); or which is basically exponential in n.

Or O(2^n);

Conditions: Basically I found condition also to check for the permutation if it is possible or not.

It will be not possible if any of the below two condition occurs.

1) For i<j<k<l and P(k)<P(j)<P(l)<P(i) .
2) For i<j<k<l and P(l)<P(j)<P(k)<P(i) .

Conclusion: If a permutation meets the above condition then, we can't generate that one using IRD. We can also check them using practical example, which is just to try each of them by trying to generate them. Which essentially I used in my code, because if I had checked that condition, it would had taken more time.

So I just tried to do it practically.