**Permutation Model**

**Time Complexity: O(n**k**).**

Since ‘generatePermutation’ is a function that is called recursively, the number of permutations depends on the user input. User inputs are: Starting letter, Ending letter, and the length of the permutation (ranging from 1 to 5)

The range from starting letter to ending letter(ascending order) or ending letter to starting letter(descending order) is denoted as ***n****.*

When starting letter is less than or equal to end, ***n* = endingLetter – startingLetter + 1.** On the other hand, if starting letter is greater than ending letter, ***n* = startingLetter –endingLetter +1.**

Ex. Starting letter = a, Ending letter = c

***n*** = c – a + 1

Referring to ASCII Value, the value of c is 99, and the value of a is 97.

***n*** = 99-97 + 1 = **3**

So there are a total of 3 characters. (**a**, **b**, **c**)

And ***k*** is the referred to as the length of the permutation defined by the user.

The recursive function iterates to generate permutation from starting letter to ending letter or ending letter to starting letter effectively making ***n*** calls for each depth of the recursion, and each call goes deeper by increasing the depth until it reaches the specified length ***k.***

Thus, the time complexity of this model is : **O(n**k**).**

Ex. Starting letter = a, Ending letter = c, Length = 2

***n*** = 99-97 + 1

***n*** = 3, ***k*** = 2

32 = **9**

Total of 9 permutations.

**Output:**

aa, ab, ac, ba, bb, bc, ca, cb, cc

**Space Complexity:** O(***k***)

The maximum depth of the recursion is ***k***, so the space complexity for the call stack is O(***k***), and the ‘result’ array takes O(***k***) as well to store current permutation. Thus, the overall time complexity is O(***k***).

Ex.

Length = 2. Therefore, the space complexity is O(2).

For visualization: For a single iteration, the 2 elements are **result[0]** and **result[1].**

|  |  |  |
| --- | --- | --- |
| result[0] | result[1] | result[2] |
| A | A | \0 (null-terminator) |

The + 1 is for the null-terminator and is omitted in space complexity. It just signifies that it is the end of the string.

**Scope of the model:**

1. The user may input any alphabet from a to z. (26 characters)
2. The user is limited to input length ranging from 1 to 5 only.
3. If the starting letter is less than the ending letter, the model will proceed in displaying output in ascending order. Otherwise, it will proceed in displaying output in descending order.