**Permutation**

Problem Statement:

Take a series of ‘n’ characters and generate all permutations.

* Input example:

Enter the String: abcd

* Output Example:

dcba

cdba

cbda

cbad

dbca

bdca

bcda

bcad

dbac

bdac

badc

bacd

dcab

cdab

cadb

cabd

dacb

adcb

acdb

acbd

dabc

adbc

abdc

abcd

Algorithm:

DEFINE FUNCTION per( p, up) :

IF (up=="") :

OUTPUT(p)

RETURN

SET ch TO up[0]

SET i TO 0

WHILE (i <= len(p)) :

SET f TO p[0:i]

SET s TO p[i:len(p)]

per(f + str(ch) + s, up[1:])

i += 1

IF \_\_name\_\_ EQUALS "\_\_main\_\_":

string=INPUT("Enter the String: ")

OUTPUT(CALL FUNCTION per("",string))

Proposed Python Code:

/\* ------- main.py ------- \*/

def per( p, up) :

if (up=="") :

print(p)

return

ch = up[0]

i = 0

while (i <= len(p)) :

f = p[0:i]

s = p[i:len(p)]

per(f + str(ch) + s, up[1:])

i += 1

if \_\_name\_\_ == "\_\_main\_\_":

string=input("Enter the String: ")

print(per("",string))

/\* ---------------------- \*/

Conclusion:

The proposed algorithm has a runtime of O(n), where n is the length of the string.

Limitations and assumptions for this algorithm include:

1.The length of the string should be in integer range.

2.The string must not contain any duplicate characters.