Alex Kitsul
230134210
CPSC 450
Assignment 5 - Report
March 23

Alex Kitsul
230134210
March 23, 2022

Psedocode

```
main:
    string <- from file

    char_list <- pre_process_string(string)
    fixed_string <- BWT(char_list)

pre_process_string(string):
    char_list <- convert string to char list
    letter_count = dictionary of letter occurance counts

    for i from 0 to len(char_list):
        update letter_count for char_list[i]
        char_list[i] = char_list[i] concat letter_count

    if no dollar sign in count, throw error
    return char_list

BWT(char_list):
    unsorted_list <- char_list
    sorted_list <- alphabetically sort char_list with $ first
    dict_list <- (keys = sorted_list, values = unsorted_list)
    new_string <- ""

    start <- "$1"
    new_string = new_string concat start[0]

    while dict_list[start] != "$1":
        new_string = new_string concat dict_list[start][0]
        start = dict_list[start]

    return reverse of new_string
```

Program Code

BurrowsWheelerTransform

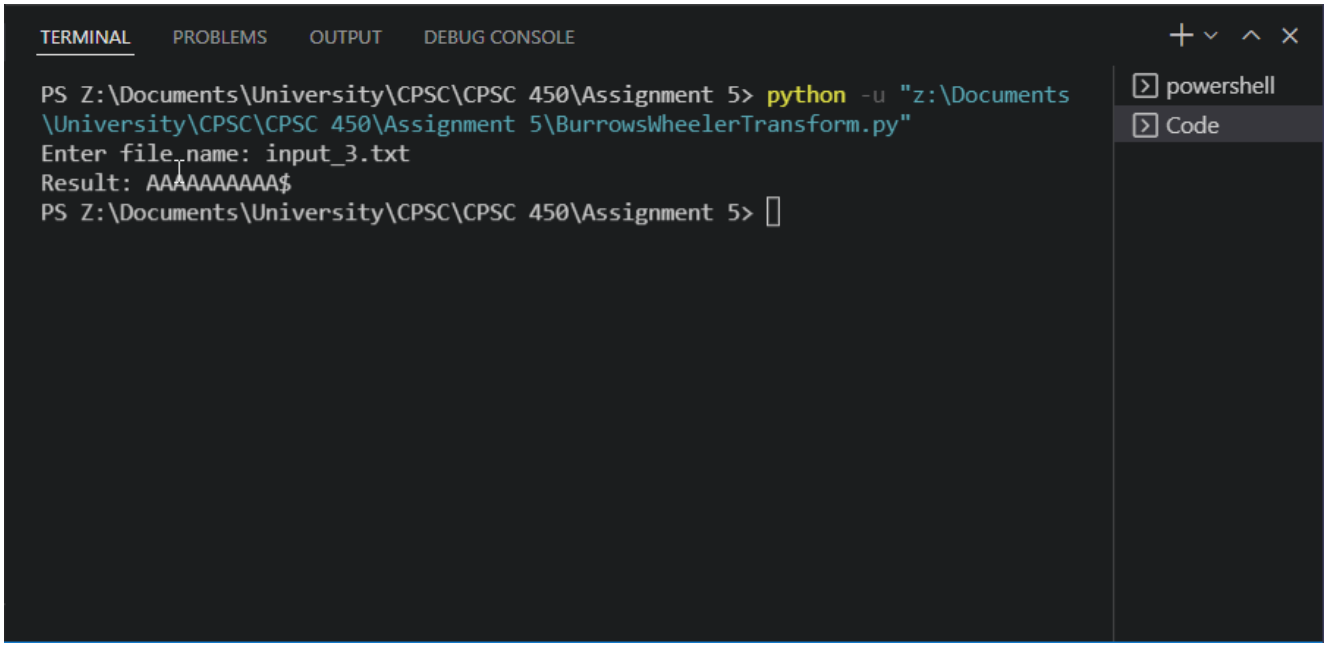
```
def main():
    """
        Main driver method

        Parameters:

            None

        Returns:

            None. Side effect.
    """
    file_name = input("Enter file name: ")

    with open(file_name) as file:
        # Read in string from file
        read_string = file.read()
        # Pre-process string by adding place numbers to the string
        char_list = pre_process_string(read_string)
        # Apply char list with new counted chars to BWT algo
        fixed_string = BWT(char_list)
        print("Result:", fixed_string)

def pre_process_string(string):
    """
        Pre processes string to count character occurances and
        splits into char array

        Parameters:

            string (String): A string with a $ somewhere in the string

        Returns:

            char_list (list[char]): A char list with counts of the occurances of
            every character ['c1', 'c2', etc...]
    """
    # Split string into char list
    char_list = list(string)
    # Make dict to store character occurance count (most O(1) way to do this)
    letter_count = dict()
    # Iterate through the char_list and replace character with itself and its
    # occurance
    for i in range(len(char_list)):
        if (char_list[i] not in letter_count):
```

```
            letter_count[char_list[i]] = 1
        else:
            letter_count[char_list[i]] += 1
        char_list[i] = char_list[i] + str(letter_count[char_list[i]])

    # If no $ was found, then string is not valid, throw exception
    if ("$" not in letter_count or letter_count["$"] > 1):
        raise ValueError("No dollar sign found in input string, \
        or too many $'s in string. Invalid.")

    return char_list

def BWT(char_list):
    """
        Execute the Inverse BTW algorithm to decode a string

        Parameters:

            char_list (list[char]): A list of characters that have been
            pre-processed by pre_process_string(string)

        Returns:

            new_string (String): The reassembled string
    """
    # Make the sorted and unsorted list into dicts to make searches
    # much more efficient
    unsorted_list = char_list
    sorted_list = sorted(char_list)
    dict_list = dict(zip(sorted_list, unsorted_list))
    new_string = ""

    # Start at the $1 and continue until you hit the $1 value
    start = "$1"
    # (Add the first $ to the string)
    new_string += start[0]
    # While we haven't hit the $1 end value, append the value (at index 0 to cut off
    # the count number, to the reassembled string)
    while (dict_list[start] != "$1"):
        new_string += dict_list[start][0]
        start = dict_list[start]

    return new_string[::-1]

if __name__ == "__main__":
    main()
```

Examples with Output

```
TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS Z:\Documents\University\CPSC\CPSC 450\Assignment 5> python -u "z:\Documents\University\CPSC\CPSC 450\Assignment 5\BurrowsWheelerTransform.py"
Enter file name: input_1.txt
Result: TACATCACGT$
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 5>
```

```
TERMINAL   PROBLEMS   OUTPUT   DEBUG CONSOLE

PS Z:\Documents\University\CPSC\CPSC 450\Assignment 5> python -u "z:\Documents\University\CPSC\CPSC 450\Assignment 5\BurrowsWheelerTransform.py"
Enter file name: input_2.txt
Result: twelveplusone$
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 5>
```


TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 5> python -u "z:\Documents\University\CPSC\CPSC 450\Assignment 5\BurrowsWheelerTransform.py"
Enter file name: input_3.txt
Result: AAAAAAAAAA$
PS Z:\Documents\University\CPSC\CPSC 450\Assignment 5>
powershell
Code