Indian Institute of Engineering Science and Technology, Shibpur

Department of Information Technology

Data Structure Lab Assignment-2

Tathagata Ghosh

2020ITB065

HY

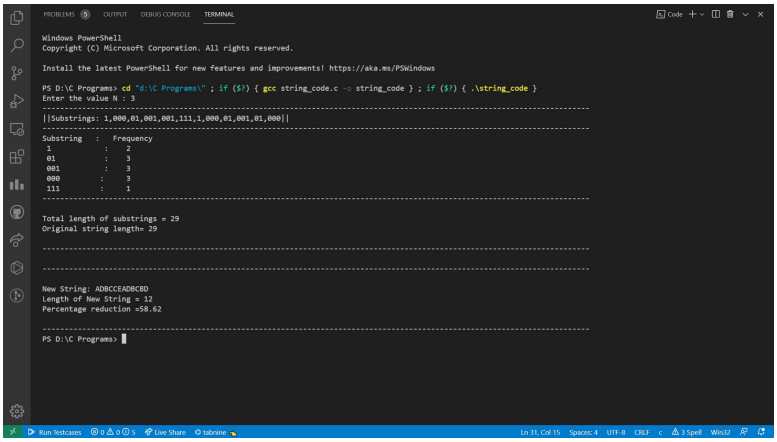
```
/*
Data Structure Lab Assignment 2021
Tathagata Ghosh --- 2020ITB065 ---- HY
16/09/2021
*/

/*
Q1. Write a program to read the string of 0s and 1s from input terminal. Then
do the
following.
a. Break the string into the substrings ending with 1 and of length of consecu
tive N
(>1) number of 0s or 1s. N can be 2, 3 and 4 taken as the input.
Example: Suppose you have a string “1000010010011111000010010101000”. For
N=4, the sub strings will be 1, 0000, 1, 001, 001, 1111, 0000,1, 001, 01, 0000
.
b. Count the frequency of each substring. Calculate the length of the string u
sing
frequency and verify with the original string length.
Example: 1 occurs 2 times, 01 occurs 1 time, 001 occurs 3 times etc.
c. Replace each pattern substring with a character starting from A (for substr
ing of
length 1), B (for substring with length 2) etc. and put the characters in a se
parate
file based on the position of the substring in the original string.
Example: 1 is replaced by A, 01 is replaced by B, 001 is replaced by C etc.
So, the final string will be: AEACCFEACBE
d. Calculate the length of the new string and calculate the % reduction of len
gth.
Example: New string is of length 11. You can now calculate the % reduction of
length.
*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define M 1000
```

```c
int main()
{
    FILE *fp;
    char ch;
    fp = fopen("test.txt", "r");
    if (fp == NULL)
    {
        printf("---File NOT Found---\n");
        exit(0);
    }
    else
    {
        char a[M];
        char res[M];
        fgets(a, M, fp);
        int l = strlen(a);

        int n;
        printf("Enter the value N : ");
        scanf("%d", &n);

        for(int i=l;i<=l+n;i++)
        {
            a[i]=a[l-1];
        }
        int freq[n+2];

        for(int i=0;i<n+2;i++)
        {
            freq[i]=0;
        }
        printf("-------------------------------------------------------------
-------------------------------------------------------------\n");
        printf("||Substrings: ");
        int i=0, cur = 0;
        while (i < l)
        {
            int j=i;
            while (j < i+n)
            {
                printf("%c", a[j]);
                j += 1;
                if (a[i] == '1')
                {
                    int pos = 1;
                    for(int k=j; k<i+n; k++)
                    {
                        if (a[k] == '0')
```

```
                        {
                            pos = 0;
                        }
                    }
                    if (!pos)
                    {
                        break;
                    }
                }
                if (a[i] == '0' && a[j-1] == '1')
                {
                    break;
                }
            }
            if (a[i] != a[j-1] || i == j-1)
            {
                freq[j-i-1] += 1;
            }
            else
            {
                freq[n+(a[i] - '0')] += 1;
            }
            res[cur] = (a[i] != a[j-1] || i==j-1) ? 'A' + (j-i-
1) : 'A' + n + (a[i] - '0');
            cur += 1;
            i = j;
            if (i < l )
            {
                printf(",");
            }
        }
        printf("||\n");
        printf("-------------------------------------------------------------
-------------------------------------------------------------\n");
        int total = 0;
        printf("Substring   :   Frequency");
        for(int i=0; i<n+2; i++)
        {
            printf("\n ");
            if (i<n)
            {
                for(int j=0; j<i; j++)
                {
                    printf("0");
                }
                printf("1");
                for(int j=0;j<n-i;j++)
                {
```

```
                    printf(" ");
                }
            }
            else
            {
                for(int j=0; j<n;j++)
                {
                    printf("%d",i-n);
                }
            }
                printf("         :\t   %d", freq[i] );
                total += freq[i]*(i<n ? i+1 : n);
        }
            printf("\n-------------------------------------------------------------
-------------------------------------------------------------------\n");
            printf("\nTotal length of substrings = %d\nOriginal string length=
 %d\n", total, l);
            printf("\n-------------------------------------------------------------
-------------------------------------------------------------------\n");
            printf("\n-------------------------------------------------------------
-------------------------------------------------------------------\n");
            printf("\nNew String: ");
            for(int i=0;i<cur; i++)
            {
                printf("%c",res[i]);
            }
            printf("\nLength of New String = %d\nPercentage reduction =%.2f\n"
, cur, (float)(total - cur) / total * 100);
            printf("\n-------------------------------------------------------------
-------------------------------------------------------------------\n");
    }
        return 0;
}
```

OUTPUT:



```c
/*
Data Structure Lab Assignment 2021
Tathagata Ghosh --- 2020ITB065 ---- HY
16/09/2021
*/

/*
Q2. Do the addition and subtraction of two NxN sparse matrices using the effic
ient
representation of the matrix done in the previous Assignment 1.
*/

#include <stdio.h>

    void add(int nz1, int nz2, int sparse1[nz1][3], int sparse2[nz2][3])
    {
        int sparse3[50][3];
        int i1 = 0, i2 = 0, i3 = 0;
        while (i1 < nz1 && i2 < nz2)
        {
            if (sparse1[i1][0] < sparse2[i2][0])
            {
                sparse3[i3][0] = sparse1[i1][0];
                sparse3[i3][1] = sparse1[i1][1];
                sparse3[i3][2] = sparse1[i1][2];
                i3++;
                i1++;
            }
            else if (sparse1[i1][0] > sparse2[i2][0])
            {
                sparse3[i3][0] = sparse2[i2][0];
                sparse3[i3][1] = sparse2[i2][1];
                sparse3[i3][2] = sparse2[i2][2];
                i3++;
                i2++;
            }
            else if (sparse1[i1][1] < sparse2[i2][1])
            {
                sparse3[i3][0] = sparse1[i1][0];
                sparse3[i3][1] = sparse1[i1][1];
                sparse3[i3][2] = sparse1[i1][2];
                i3++;
                i1++;
            }
            else if (sparse1[i1][1] > sparse2[i2][1])
            {
                sparse3[i3][0] = sparse2[i2][0];
                sparse3[i3][1] = sparse2[i2][1];
```

```
                sparse3[i3][2] = sparse2[i2][2];
                i3++;
                i2++;
            }
            else
            {
                if (sparse1[i1][2] + sparse2[i2][2] != 0)
                {
                    sparse3[i3][0] = sparse1[i1][0];
                    sparse3[i3][1] = sparse1[i1][1];
                    sparse3[i3][2] = sparse1[i1][2] + sparse2[i2][2];
                    i1++;
                    i2++;
                    i3++;
                }
                else
                {
                    i1++;
                    i2++;
                }
            }
        }
        while (i1 < nz1)
        {
            sparse3[i3][0] = sparse1[i1][0];
            sparse3[i3][1] = sparse1[i1][1];
            sparse3[i3][2] = sparse1[i1][2];
            i3++;
            i1++;
        }
        while (i2 < nz2)
        {
            sparse3[i3][0] = sparse2[i2][0];
            sparse3[i3][1] = sparse2[i2][1];
            sparse3[i3][2] = sparse2[i2][2];
            i3++;
            i2++;
        }
        printf("Sparse Matrix after Addition:\n");
        printf("row\tcol\tvalue\n");
        for (int i = 0; i < i3; i++)
        {
            printf("%d\t%d\t%d\n", sparse3[i][0], sparse3[i][1], sparse3[i][2]
);
        }
    }

    void subtract(int nz1, int nz2, int sparse1[nz1][3], int sparse2[nz2][3])
```

```
{
    int sparse3[50][3];
    int i1 = 0, i2 = 0, i3 = 0;
    while (i1 < nz1 && i2 < nz2)
    {
        if (sparse1[i1][0] < sparse2[i2][0])
        {
            sparse3[i3][0] = sparse1[i1][0];
            sparse3[i3][1] = sparse1[i1][1];
            sparse3[i3][2] = sparse1[i1][2];
            i3++;
            i1++;
        }
        else if (sparse1[i1][0] > sparse2[i2][0])
        {
            sparse3[i3][0] = sparse2[i2][0];
            sparse3[i3][1] = sparse2[i2][1];
            sparse3[i3][2] = -sparse2[i2][2];
            i3++;
            i2++;
        }
        else if (sparse1[i1][1] < sparse2[i2][1])
        {
            sparse3[i3][0] = sparse1[i1][0];
            sparse3[i3][1] = sparse1[i1][1];
            sparse3[i3][2] = sparse1[i1][2];
            i3++;
            i1++;
        }
        else if (sparse1[i1][1] > sparse2[i2][1])
        {
            sparse3[i3][0] = sparse2[i2][0];
            sparse3[i3][1] = sparse2[i2][1];
            sparse3[i3][2] = -sparse2[i2][2];
            i3++;
            i2++;
        }
        else
        {
            if (sparse1[i1][2]  != sparse2[i2][2])
            {
                sparse3[i3][0] = sparse1[i1][0];
                sparse3[i3][1] = sparse1[i1][1];
                sparse3[i3][2] = sparse1[i1][2] - sparse2[i2][2];
                i1++;
                i2++;
                i3++;
            }
```

```
                else
                {
                    i1++;
                    i2++;
                }
            }

        }
        while (i1 < nz1)
        {
            sparse3[i3][0] = sparse1[i1][0];
            sparse3[i3][1] = sparse1[i1][1];
            sparse3[i3][2] = sparse1[i1][2];
            i3++;
            i1++;
        }
        while (i2 < nz2)
        {
            sparse3[i3][0] = sparse2[i2][0];
            sparse3[i3][1] = sparse2[i2][1];
            sparse3[i3][2] = -sparse2[i2][2];
            i3++;
            i2++;
        }

        printf("Sparse Matrix after Subtraction:\n");
        printf("row\tcol\tvalue\n");
        for (int i = 0; i < i3; i++)
        {
            printf("%d\t%d\t%d\n", sparse3[i][0], sparse3[i][1], sparse3[i][2]
);
        }
    }

    int main()
    {
        int n = 5;
        printf("Enter the dimension of matrix 1: ");
        scanf("%d", &n);
        int arr1[n][n];
        int nonzero1 = 0;
        printf("Enter the elements of matrix 1: \n");
        for (int i = 0; i < n; i++)
        {
            for (int j = 0; j < n; j++)
            {
                scanf("%d", &arr1[i][j]);
                if (arr1[i][j] != 0)
```

```
            {
                nonzero1++;
            }
        }
    }

    int arr2[n][n];
    int nonzero2 = 0;
    printf("Enter the elements of matrix 2: \n");
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            scanf("%d", &arr2[i][j]);
            if (arr2[i][j] != 0)
            {
                nonzero2++;
            }
        }
    }

    int sparse1[nonzero1][3];
    int k = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (arr1[i][j] != 0)
            {
                sparse1[k][0] = i;
                sparse1[k][1] = j;
                sparse1[k][2] = arr1[i][j];
                k++;
            }
            if (k == nonzero1)
            {
                break;
            }
        }
    }

    int sparse2[nonzero2][3];
    k = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (arr2[i][j] != 0)
```

```
                {
                    sparse2[k][0] = i;
                    sparse2[k][1] = j;
                    sparse2[k][2] = arr2[i][j];
                    k++;
                }
                if (k == nonzero2)
                {
                    break;
                }
            }
        }

        printf("Sparse Matrix Representation of 1st Matrix: \n");
        printf("row\tcol\tvalue\n");

        for (int i = 0; i < nonzero1; i++)
        {
            printf("%d\t%d\t%d\n", sparse1[i][0], sparse1[i][1], sparse1[i][2]
);
        }

        printf("Sparse Matrix Representation of 2nd Matrix: \n");
        printf("row\tcol\tvalue\n");

        for (int i = 0; i < nonzero2; i++)
        {
            printf("%d\t%d\t%d\n", sparse2[i][0], sparse2[i][1], sparse2[i][2]
);
        }

        int x;

        printf("Enter 1 for addition and 2 for subtraction: \n");
        scanf("%d", &x);

        switch(x)
        {
            case 1: add(nonzero1, nonzero2, sparse1, sparse2);
                    break;
            case 2: subtract(nonzero1, nonzero2, sparse1, sparse2);
                    break;
            default: printf("Enter a valid choice!");
        }
        return 0;
    }
```

OUTPUT:-

```
PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL

--------------------------------------------------------------------------------------------------------------------------------
PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc sparse_add_sub.c -o sparse_add_sub } ; if ($?) { .\sparse_add_sub }
Enter the dimension of matrix 1: 3
Enter the elements of matrix 1:
1 0 0
0 1 0
0 0 1
Enter the elements of matrix 2:
0 0 1
0 1 0
1 0 0
Sparse Matrix Representation of 1st Matrix:
row     col     value
0       0       1
1       1       1
2       2       1
Sparse Matrix Representation of 2nd Matrix:
row     col     value
0       2       1
1
Sparse Matrix after Addition:
row     col     value
0       0       1
0       2       1
1       1       2
2       0       1
2       2       1
PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc sparse_add_sub.c -o sparse_add_sub } ; if ($?) { .\sparse_add_sub }
Enter the dimension of matrix 1: 3
Enter the elements of matrix 1:
0 0 -2
0 1 0
0 0 0
Enter the elements of matrix 2:
1 0 2
0 0 0
0 0 5
Sparse Matrix Representation of 1st Matrix:
row     col     value
```

```
PROBLEMS 5   OUTPUT   DEBUG CONSOLE   TERMINAL

1
Sparse Matrix after Addition:
row     col     value
0       0       1
0       2       1
1       1       2
2       0       1
2       2       1
PS D:\C Programs> cd "d:\C Programs\" ; if ($?) { gcc sparse_add_sub.c -o sparse_add_sub } ; if ($?) { .\sparse_add_sub }
Enter the dimension of matrix 1: 3
Enter the elements of matrix 1:
0 0 -2
0 1 0
0 0 0
Enter the elements of matrix 2:
1 0 2
0 0 0
0 0 5
Sparse Matrix Representation of 1st Matrix:
row     col     value
0       2       -2
1       1       1
Sparse Matrix Representation of 2nd Matrix:
row     col     value
0       0       1
0       2       2
2       2       5
Enter 1 for addition and 2 for subtraction:
2
Sparse Matrix after Subtraction:
row     col     value
0       0       -1
0       2       -4
1       1       1
2       2       -5
PS D:\C Programs>
```

GitHub: https://github.com/Tathagata-Ghosh-Developer/Lab-Assignment-3rd-Semester