Exchanges

Shashwat Gupta (14IE10028)

March 28, 2016

1 Input of Table

I have created an universal 2D array storing the conversion rates. I have used the 3X3 data given in the question.

The following is the code for my data used in the program

```
int tread[N][N] = {{100, 200, 400}, {50, 100, 220}, {25, 45, 100}};
```

Listing 1: Input

The user can also enter his own data by changing the N in the hash define and also uncommenting the commented lines in the following void read()

```
void read()
{
    /* int i,j;
    for(i=0; i<N; ++i)
    {
        for(j=0; j<N; ++j)
            scanf(”%d”, &tread[i][j]);
    }
    putchar('\n'); */
    printf(”Convertion Matrix:\n”);
    printMatrix(tread);
}
```

Listing 2: Input

2 Calculation of path costs

The following is the code which calculates the path costs for the graph and stores it in a temporary array and then prints it.

My method is Floyd Warshall. The modification made is to maximise the products of the new path route to maximise the profits. This product is compared with the initial value and the larger is chosen automatically to get the desired output of maximum value attainable.

```
int find_profit()
{
    double temp[N][N];
    int i,j,k;
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
            temp[i][j]=tread[i][j];
    }
    for(k=0; k<N; ++k)
    {
        for(i=0; i<N; ++i)
        {
            for(j=0; j<N; ++j)
            {
                if(temp[i][k]*temp[k][j]/temp[i][i] > temp[i][j])
                    temp[i][j] = temp[i][k]*temp[k][j]/temp[i][i];
            }
        }
    }
    printf("Max Sum Matrix:\n");
    printMatrix(temp);
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
            temp[i][j]=temp[i][j]-tread[i][j];
    }
    printf("Profit Matrix:\n");
    printMatrix(temp);
    for (i = 0; i < N; ++i)
    {
        if(temp[i][i]>0)
            return 1;
    }
    return 0;
}
```

Listing 3: read

3 Printing

The following is the code which prints the 2D matrix.

```
void printMatrix(double a[][N])
{
    int i,j;
    for (i = 0; i < N; ++i)
    {
        for (j = 0; j < N; ++j)
        printf("%f\t", a[i][j]);
        printf("\n");
    }
    printf("\n");
}
```

Listing 4: print

4 Complexity

These are the costs:

1. Insert $\in O(n^2)$.
2. Calculate cost $\in O(n^3)$.
3. Find Path possibility $\in O(n)$.
4. Printing $\in O(n^2)$.