Name : Prem Bansod

Roll no. 41310

Subject : LP2

Assignment OR 1

Code:

```
import java.util.*;
class Transportation
{
   int[][] cost;
   int[] supply;
   int[] demand;
   public Transportation()
   {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter number of suppliers");
      int s = sc.nextInt();
      System.out.println("Enter number of demands");
      int d = sc.nextInt();
      supply = new int[s];
      demand = new int[d];
      cost = new int[s][d];
   }
   public void input()
   {
      Scanner sc = new Scanner(System.in);
      for(int i = 0;i<supply.length;i++)
      {
         System.out.println("Enter supply of supplier = "+i);
         supply[i] = sc.nextInt();
      }
      for(int i = 0;i<demand.length;i++)
      {
         System.out.println("Enter demand of "+i);
         demand[i] = sc.nextInt();
      }
      int n = cost.length;
      int m = cost[0].length;
      System.out.println("Enter cost matrix");
      for(int i = 0;i<n;i++)
      {
         for(int j = 0;j<m;j++)
         {
            cost[i][j] = sc.nextInt();
         }
      }
   }
   public void printData()
```

```
{
    int n = cost.length;
    int m = cost[0].length;
    System.out.println("Cost matrix");
    for(int i= 0;i<n;i++)
    {
        for(int j = 0;j<m;j++)
        {
            System.out.print(cost[i][j]+" ");
        }
        System.out.println();
    }
    System.out.print("Supply = ");
    for(int i = 0;i<n;i++)
    {
        System.out.print(supply[i]+" ");
    }
    System.out.print("\nDemand = ");
    for(int i = 0;i<m;i++)
    {
        System.out.print(demand[i]+" ");
    }
    System.out.println();
}
public boolean check()
{
    int s = 0;
    int d = 0;
    for(int i = 0;i<supply.length;i++)
    {
        s = s + supply[i];
    }
    for(int i = 0;i<demand.length;i++)
    {
        d = d + demand[i];
    }
    if(s == d)
    {
        return true;
    }
    else
    {
        return false;
    }
}
public int northWestCornerCell()
{
    int res = 0;
    int n = supply.length;
    int m = demand.length;
    if(check())
    {
```

```
        int[] supply = new int[n];
        for(int i = 0;i<n;i++)
        {
           supply[i] = this.supply[i];
        }
        int[] demand = new int[m];
        for(int i = 0;i<m;i++)
        {
           demand[i] = this.demand[i];
        }
        System.out.println("Balanced problem");
        int i = 0;
        int j = 0;

        while(i<n&&j<m)
        {
              if(supply[i]<demand[j])
              {
                 demand[j] = demand[j] - supply[i];
                 res = res + cost[i][j] * supply[i];
                 supply[i] = 0;
                 i++;
              }
              else
              {
                 supply[i] = supply[i] - demand[j];
                 res = res + cost[i][j] * demand[j];
                 demand[j] = 0;
                 j++;
              }
        }
     }
     else
     {
        System.out.println("Imbalanced Problem");
     }
     return res;
}
public int leastCostCell()
{
     int res = 0;
     int n = supply.length;
     int m = demand.length;
     if(check())
     {
        int[] supply = new int[n];
        for(int i = 0;i<n;i++)
        {
           supply[i] = this.supply[i];
        }
        int[] demand = new int[m];
        for(int i = 0;i<m;i++)
```

```
    {
       demand[i] = this.demand[i];
    }
    System.out.println("Balanced Problem");
    boolean[] s = new boolean[n];
    boolean[] d = new boolean[m];
    int i = 0,j = 0;
    int tempi = 0,tempj = 0;
    while(true)
    {
       int min = Integer.MAX_VALUE;
       for(i = 0;i<n;i++)
       {
          for(j = 0;j<m;j++)
          {
             if(s[i] ==false && d[j] == false)
             {
                if(cost[i][j]<min)
                {
                   min = cost[i][j];
                   tempi = i;
                   tempj = j;
                }
             }
             else if (s[i] == true)
             {
                break;
             }
          }
       }
//    System.out.println("Minimum = "+min + " i = "+tempi+" j = "+tempj);
      if(supply[tempi]<demand[tempj] && !s[tempi])
       {
          res = res + cost[tempi][tempj] * supply[tempi];
          s[tempi] = true;
          demand[tempj] = demand[tempj] - supply[tempi];
       }
       else if(!d[tempj])
       {
          res = res + cost[tempi][tempj] * demand[tempj];
          d[tempj] = true;
          supply[tempi] = supply[tempi] - demand[tempj];
       }
       for(i = 0;i<n;i++)
       {
          if(!s[i])
          {
             break;
          }
       }
       for(j = 0;j<m;j++)
       {
```

```
                if(!d[j])
                {
                    break;
                }
            }
            if(i == n||j == m)
            {
                break;
            }
        }
    }
    else
    {
        System.out.println("Unbalanced Problem");
    }
    return res;
}
public int vogels()
{
    int res = 0;
    int n = supply.length;
    int m = demand.length;
    if(check())
    {
        int[] supply = new int[n];
        for(int i = 0;i<n;i++)
        {
            supply[i] = this.supply[i];
        }
        int[] demand = new int[m];
        for(int i = 0;i<m;i++)
        {
            demand[i] = this.demand[i];
        }
        System.out.println("Balanced Problem");
        boolean[] s = new boolean[n];
        boolean[] d = new boolean[m];
        int i = 0,j = 0;
        int tempi = 0,tempj = 0;
        while(true)
        {
            int min1 = Integer.MAX_VALUE;
            int min2 = 0;
            int ri = 0,rj = 0;
            int diff = Integer.MIN_VALUE;
            for(i = 0;i<n;i++)
            {
                min1 = Integer.MAX_VALUE;
                for(j = 0;j<m;j++)
                {
                    if(s[i] ==false && d[j] == false)
                    {
```

```
            if(cost[i][j]<min1)
            {
               min2 = min1;
               min1 = cost[i][j];
               ri = i;
               rj = j;
            }
            else if(cost[i][j]>min1&&cost[i][j]<min2)
            {
               min2 = cost[i][j];
            }
         }
         else if (s[i] == true)
         {
            break;
         }
      }
      if(!s[i])
      {
         if(min2 == Integer.MAX_VALUE)
         {
            min2 = 0;
         }
         if(diff<Math.abs(min1-min2))
         {
            tempi = ri;
            tempj = rj;
            diff = Math.abs(min1-min2);
         }
      }
     //  System.out.println("Diff = "+diff+"min1 = "+min1+"min2 = "+min2);
   }
  // System.out.println("cols");
   min1 = Integer.MAX_VALUE;
   min2 = 0;
   for(i = 0;i<m;i++)
   {
      min1 = Integer.MAX_VALUE;
      for(j = 0;j<n;j++)
      {
         if(s[j] ==false && d[i] == false)
         {
            if(cost[j][i]<min1)
            {
               min2 = min1;
               min1 = cost[j][i];
               ri = j;
               rj = i;
            }
            else if(cost[j][i]>min1&&cost[j][i]<min2)
            {
               min2 = cost[j][i];
```

```
            }
        }
        else if (d[i] == true)
        {
            break;
        }
    }
    if(!d[i])
    {
        if(min2 == Integer.MAX_VALUE)
        {
            min2 = 0;
        }
        if(diff<Math.abs(min1-min2))
        {
            tempi = ri;
            tempj = rj;
            diff = Math.abs(min1-min2);
        }
    }
  //   System.out.println("Diff = "+diff+"min1 = "+min1+"min2 = "+min2);
  }
 // System.out.println("Minimum = "+cost[tempi][tempj] + " i = "+tempi+" j = "+tempj);
if(supply[tempi]<demand[tempj] && !s[tempi])
  {
    res = res + cost[tempi][tempj] * supply[tempi];
    s[tempi] = true;
    demand[tempj] = demand[tempj] - supply[tempi];
  }
  else if(!d[tempj])
  {
    res = res + cost[tempi][tempj] * demand[tempj];
    d[tempj] = true;
    supply[tempi] = supply[tempi] - demand[tempj];
  }
  for(i = 0;i<n;i++)
  {
    if(!s[i])
    {
        break;
    }
  }
  for(j = 0;j<m;j++)
  {
    if(!d[j])
    {
        break;
    }
  }
  if(i == n||j == m)
  {
    break;
```

```
            }
          }
        }
        else
        {
          System.out.println("Unbalanced Problem");
        }
        return res;
    }
    public static void main(String[] args)
    {
        Transportation obj = new Transportation();
        obj.input();
        obj.printData();
        System.out.println("Minimum cost By North-West Corner Cell =
"+obj.northWestCornerCell());
        System.out.println("Minimum cost By Least Cost Cell = "+obj.leastCostCell());
        System.out.println("Minimum cost By Vogel's Approximation Method = "+obj.vogels());
    }
}
```

Output :

```
Enter number of suppliers
3
Enter number of demands
4
Enter supply of supplier = 0
7
Enter supply of supplier = 1
9
Enter supply of supplier = 2
18
Enter demand of 0
5
Enter demand of 1
8
Enter demand of 2
7
Enter demand of 3
14
Enter cost matrix
19 30 50 10
70
30 40 60
40 8 70 20
Cost matrix
19 30 50 10
70 30 40 60
40 8 70 20
Supply = 7 9 18
```

Demand = 5 8 7 14
Balanced problem
Minimum cost By North-West Corner Cell = 1015
Balanced Problem
Minimum cost By Least Cost Cell = 814
Balanced Problem
Minimum cost By Vogel's Approximation Method = 779