DWM3-1211061

Aim : Implementation of Apriori association rule mining algorithm

Problem Statement:

No.	I1 Nominal	I2 Nominal	I3 Nominal	I4 Nominal	I5 Nominal
1	yes	yes			yes
2		yes		yes	
3		yes	yes		
4	yes	yes		yes	
5	yes		yes		
6		yes	yes		
7	yes		yes		
8	yes	yes	yes		yes
9	yes	yes	yes		

Code:

```
//apriori.java
import java.util.*;
import java.io.*;

class apriori
{
	static int comb (int n,int r)
	{
		if (r==0||r==n)
			return 1;
		else
			return comb(n-1,r-1)+comb(n-1,r);
	}


	public static void main(String args[])throws IOException
	{
		BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
		int a[][] = new int[9][6];
		int l1[] = new int[5];
		int count[] = new int[5];
		int c=0;
		for(int k=0;k<5;k++)
		{
			count[k] = 0;
			l1[k] = 0;
		}
		for(int i=0;i<9;i++)
		{
			for(int j=0;j<6;j++)
			{
				a[i][j] = Integer.parseInt(br.readLine());
			}
		}
		System.out.println();
		System.out.println("Input:");
		for(int i=0;i<9;i++)
		{
			for(int j=0;j<6;j++)
```

```
		{
			System.out.print(a[i][j]+" ");
		}
		System.out.println();
}
System.out.println();
int k=0;
for(int j=1;j<6;j++)
{
		for(int i=0;i<9;i++)
		{
			count[k]+=a[i][j];
		}
		k++;
}


System.out.print("C1 = ");
for(k=0;k<5;k++)
{
		System.out.print("I"+(k+1)+" ");
}
System.out.println();
System.out.print("     ");
for(k=0;k<5;k++)
{
		System.out.print(count[k]+"  ");
}
double sup=Double.parseDouble(br.readLine());
System.out.println();
sup=Math.ceil((sup/100.0)*9);
System.out.println();
System.out.println("Support = "+sup);
System.out.println();
int l1flag=0;
System.out.print("L1 = ");
int l1c=0,z=1;
for(k=0;k<5;k++)
{
		if(count[k]>=sup)
```

```
		{
			l1flag=1;
			System.out.print("I"+(k+1)+" ");
			l1c++;
		}
	}
	for(k=0;k<l1c;k++)
	{
		if(count[k]>=sup)
		{
			l1[c++]=k+1;
		}
	}
	if(l1flag==0)
	{
		System.out.println("Empty!");
		return;
	}
	System.out.println();
	System.out.print("     ");
	for(k=0;k<l1c;k++)
	{
		if(l1c==1)
			System.out.print(count[1]+"  ");
		else
			System.out.print(count[l1[k]-1]+"  ");
	}
	System.out.println();
	System.out.println();
	if(c<2)
		return;
	System.out.println("No. of elements: "+c);
	int comb1=comb(c,2);
	System.out.println();
	System.out.println("No. of pairs: "+comb1);

	int c2[] = new int[2*comb1];
	int x=0;

	for(int i=0;i<c;i++)
```

```
{
        for(int j=i+1;j<c;j++)
        {
                c2[x++]=l1[i];
                c2[x++]=l1[j];
        }
}
System.out.println();
System.out.println("All Combinations:");
System.out.print("C2 = ");
for(k=0;k<2*comb1;k=k+2)
{
        System.out.print("I"+c2[k]+"I"+c2[k+1]+" ");
}

int c2c[]=new int[comb1];
int p=0;
for(k=0;k<comb1;k++)
{
        c2c[k]=0;
}
for(int j=0;j<2*comb1;j=j+2)
{
        for(int i=0;i<9;i++)
        {
                if(a[i][c2[j]]*a[i][c2[j+1]]==1)
                        c2c[p]++;
        }
        p++;
}
System.out.println();
System.out.print("      ");
for(k=0;k<comb1;k++)
{
        System.out.print(c2c[k]+"    ");
}
int t=0;
System.out.println();

int l2c=0;
```

```
System.out.println();
int l2flag=0;
System.out.print("L2 = ");
for (int i=0;i<comb1;i++)
{
	if(c2c[i]>=sup)
	{
		l2c++;
		System.out.print("I"+(c2[2*i])+"I"+(c2[2*i+1])+" ");
		l2flag=1;
	}
	/*else
	{
		c2[2*i]=c2[2*i+1]=0;
	}*/
}
if(l2flag==0)
{
	System.out.println("Empty!");
	return;
}
System.out.println();
System.out.print("      ");
int l2[]=new int[l2c];
int q=0;
for (int i=0;i<comb1;i++)
{
	if(c2c[i]>=sup)
	{
		l2[q++]=c2c[i];
		System.out.print(l2[q-1]+"    ");
	}
}
System.out.println();
int l2f[]=new int[2*l2c];
int l=0;
for (int i=0;i<comb1;i++)
{
	if(c2c[i]>=sup)
	{
```

```
                l2f[l++]=c2[2*i];
                l2f[l++]=c2[2*i+1];
                //System.out.print(l2f[l-2]+" "+l2f[l-1]+" ");
        }
}

System.out.println();

int subs[][]=new int[5][5];
for(int i=0;i<5;i++)
{
        for(int j=0;j<5;j++)
        {
                subs[i][j]=0;
        }
}


for(k=0;k<l;k=k+2)
{
        subs[l2f[k]-1][l2f[k+1]-1]=1;
}

//System.out.println(l/2);
int c3[]=new int[l/2];
int c3c[]=new int[20];
for(k=0;k<l/2;k++)
{
        c3[k]=0;
}
for(k=0;k<20;k++)
{
        c3c[k]=0;
}
int r=0,s=0,l3flag=0;
System.out.println("C3 (First 1 symbol same) = ");
for (int i=0;i<2*l2c;i=i+2)
{
        for (int j=i+2;j<2*l2c;j=j+2)
        {
```

```
                    if(l2f[i]==l2f[j])
                    {
                            System.out.print("I"+l2f[i]+"I"+l2f[i+1]+"I"+l2f[j+1]+" ");

                            for(k=0;k<9;k++)
                            {
                                    if(a[k][l2f[i]]*a[k][l2f[i+1]]*a[k][l2f[j+1]]==1)
                                    {
                                            c3[r]++;
                                    }
                            }
                            r++;
                            if(c3[r-1]>=sup)
                            {
                                    l3flag=1;
                                    c3c[s++]=l2f[i];
                                    c3c[s++]=l2f[i+1];
                                    c3c[s++]=l2f[j+1];
                            }
                    }
            }
    }


    System.out.println();

    for(k=0;k<l/2;k++)
    {
            System.out.print("  "+c3[k]+"    ");
    }

    System.out.println();
    System.out.println();
    System.out.println("Before checking for subsets: L3 = ");
    if(l3flag==0)
    {
            System.out.println("Empty!");
            return;
    }
    for(k=0;k<s;k=k+3)
```

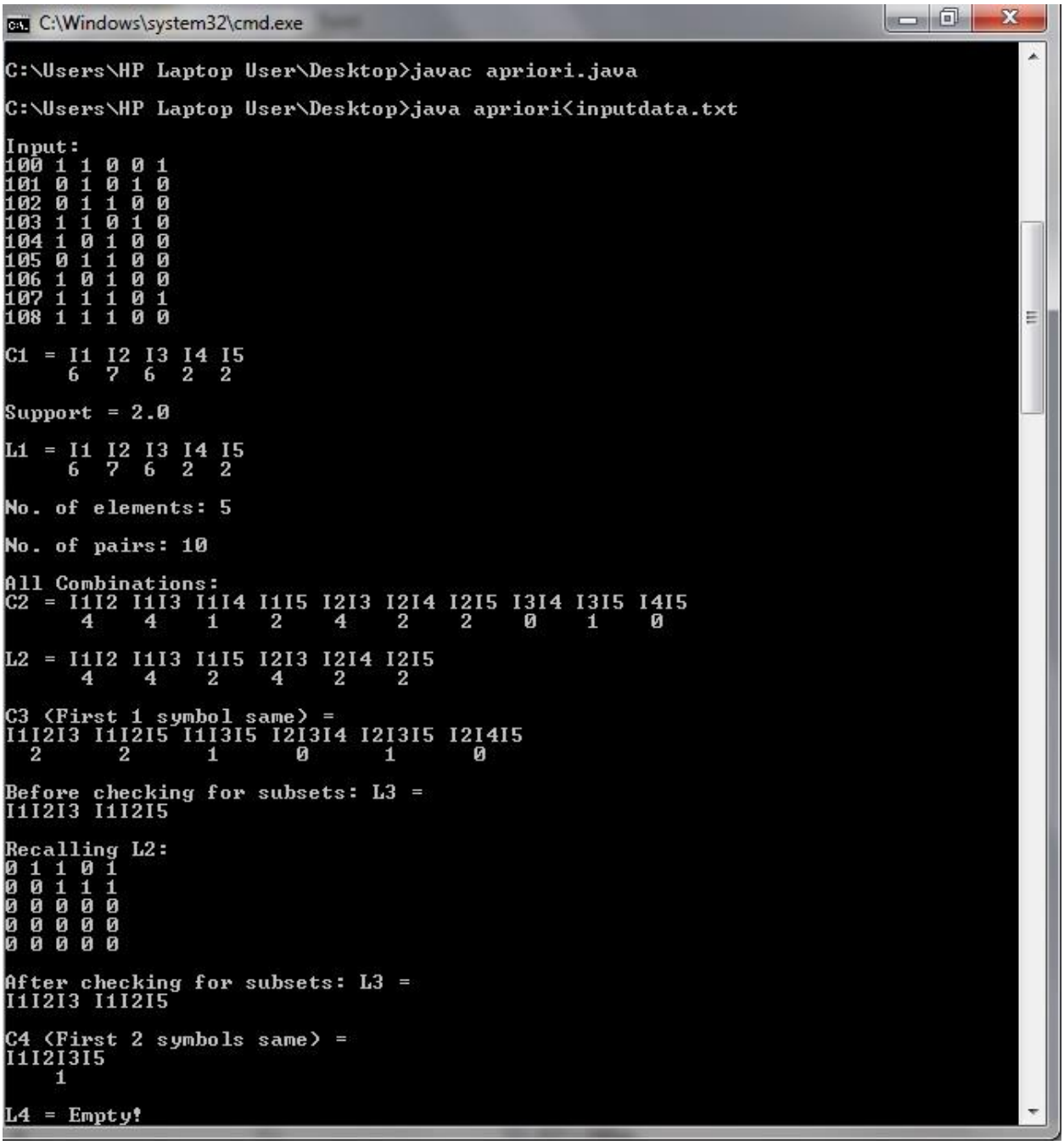
```
		{
			System.out.print("I"+c3c[k]+"I"+c3c[k+1]+"I"+c3c[k+2]+" ");
		}

		System.out.println();
		System.out.println();
		System.out.println("Recalling L2:");
		for(int i=0;i<5;i++)
		{
			for(int j=0;j<5;j++)
			{
				System.out.print(subs[i][j]+" ");
			}
			System.out.println();
		}

		// check for all subsets

		int fc3[]=new int[s];
		for(k=0;k<s;k++)
		{
			fc3[k]=c3c[k];
		}

		//System.out.println(s);
		int fc3c=0,flag2=0;
		for(k=0;k<s;k=k+3)
		{
			//System.out.println(subs[c3c[k]-1][c3c[k+1]-1]+" "+subs[c3c[k]-
1][c3c[k+2]-1]+" "+subs[c3c[k+1]-1][c3c[k+2]-1]);
			if(subs[c3c[k]-1][c3c[k+1]-1]*subs[c3c[k]-1][c3c[k+2]-1]*subs[c3c[k+1]-
1][c3c[k+2]-1]==0)
				c3c[k]=c3c[k+1]=c3c[k+2]=0;

		}
		/*for(k=0;k<s;k++)
		{
			System.out.print(c3c[k]+" ");
		}*/
		for(k=0;k<s;k++)
```

```
{
	if(c3c[k]!=0)
	{
		fc3[fc3c++]=c3c[k];
		flag2=1;
	}
}
System.out.println();

System.out.println("After checking for subsets: L3 = ");

if(flag2==0)
{
	System.out.println("Empty!");
	return;
}
for(k=0;k<fc3c;k=k+3)
{
	System.out.print("I"+fc3[k]+"I"+fc3[k+1]+"I"+fc3[k+2]+" ");
}

/*int fc3f[] = new int[k];
for(int i=0;i<k;i++)
{
	fc3f[i]=fc3[i];
}*/

int l3[] = new int[10];
int c4[] = new int[5];
int l3c=0,c4c=0;
System.out.println();
System.out.println();
System.out.println("C4 (First 2 symbols same) = ");
for(k=0;k<fc3c;k=k+3)
{
	for(int i=k+3;i<fc3c;i=i+3)
	{
		if(fc3[k]==fc3[i]&&fc3[k+1]==fc3[i+1])
		{
			l3[l3c++]=fc3[k];
```

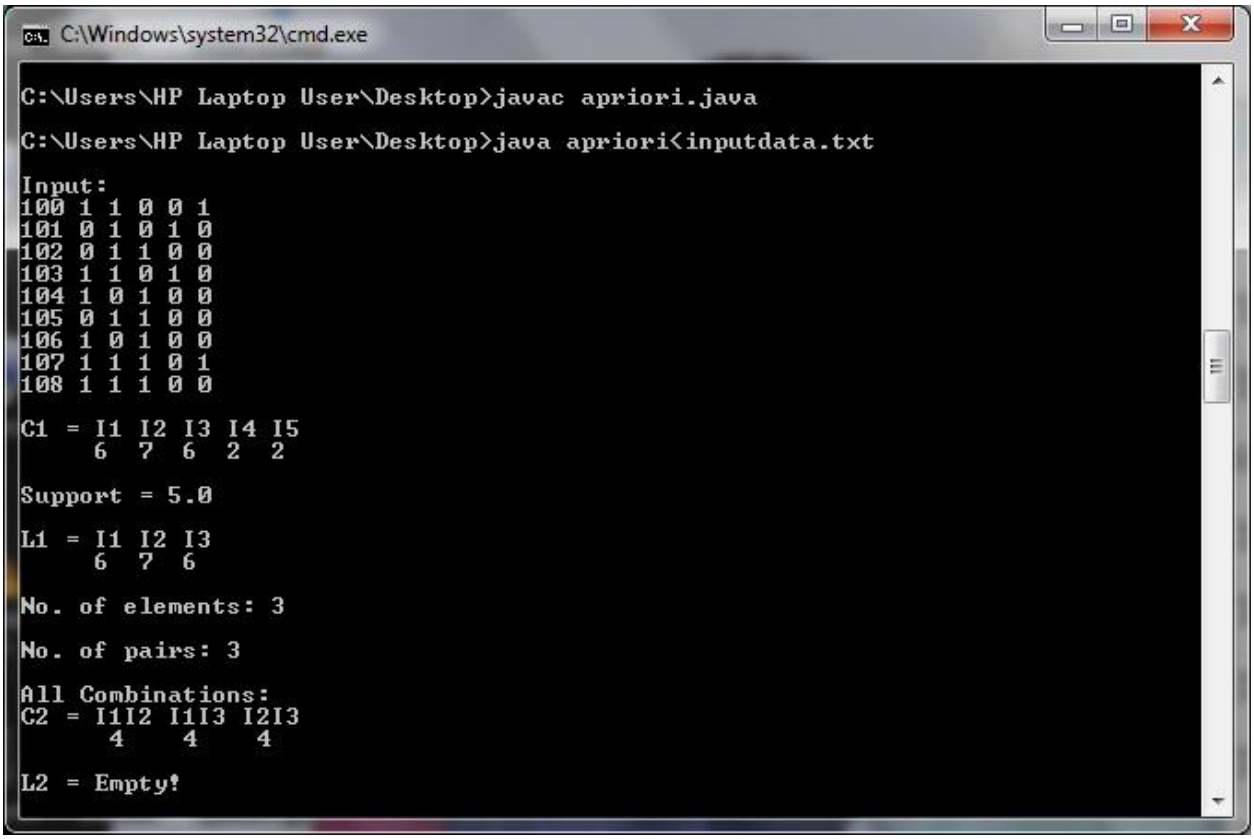
```
                                        l3[l3c++]=fc3[k+1];
                                        l3[l3c++]=fc3[k+2];
                                        l3[l3c++]=fc3[i+2];
                                        for(int j=0;j<9;j++)
                                        {

        if(a[j][fc3[k]]*a[j][fc3[k+1]]*a[j][fc3[k+2]]*a[j][fc3[i+2]]==1)
                                                {
                                                        c4[c4c]++;
                                                }
                                        }
                                        c4c++;
                                }
                        }
                }
                for(k=0;k<l3c/4;k++)
                        System.out.print("I"+l3[k]+"I"+l3[k+1]+"I"+l3[k+2]+"I"+l3[k+3]+" ");
                System.out.println();
                System.out.print("    ");
                for(k=0;k<c4c;k++)
                        System.out.print(c4[k]+"    ");
                System.out.println();
                System.out.println();
                int l4flag=0;
                System.out.print("L4 = ");
                for(k=0;k<c4c;k=k+4)
                        if(c4[k]>=sup)
                        {

        System.out.print("I"+l3[k]+"I"+l3[k+1]+"I"+l3[k+2]+"I"+l3[k+3]+" ");
                                l4flag=1;
                        }
                if(l4flag==0)
                        System.out.println("Empty!");



        }
}
```

Output:

Support = 22%



Support = 50%


C:\Windows\system32\cmd.exe
C:\Users\HP Laptop User\Desktop>javac apriori.java
C:\Users\HP Laptop User\Desktop>java apriori<inputdata.txt
Input:
100 1 1 0 0 1
101 0 1 0 1 0
102 0 1 1 0 0
103 1 1 0 1 0
104 1 0 1 0 0
105 0 1 1 0 0
106 1 0 1 0 0
107 1 1 1 0 1
108 1 1 1 0 0
C1 = I1 I2 I3 I4 I5
6 7 6 2 2
Support = 5.0
L1 = I1 I2 I3
6 7 6
No. of elements: 3
No. of pairs: 3
All Combinations:
C2 = I1I2 I1I3 I2I3
4 4 4
L2 = Empty!