

COMP 610 MAX2SAT PROJECT CODE

HARISH PINDI

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class HarishMax2Sat {

    public static void quick(ArrayList arr) {

        ArrayList distArr = getDistinct(arr);
        String tru[] = new String[distArr.size()];
        for (int i = 0; i < distArr.size(); i++) {
            int countPos = 0;
            int countNeg = 0;
            int x = Integer.parseInt(distArr.get(i).toString());
            for (int j = 0; j < arr.size(); j = j + 2) {
                int y = Integer.parseInt(arr.get(j).toString());
                int z = Integer.parseInt(arr.get(j + 1).toString());
                if (x == y) {
                    countPos++;
                } else if (x == z) {
                    countPos++;
                } else if (x == -y) {
```

```

countNeg++;
} else if (x == -z) {
countNeg++;
}
}

if (countPos > countNeg) {
tru[i] = "1";
} else {
tru[i] = "-1";
}
}

for (int i = 0; i < tru.length; i++) {
if(tru[i].equals("1")){
System.out.print("T ");
}else{
System.out.print("F ");
}
}

System.out.print(max2sat(arr, distArr, tru)+" ");
System.out.println();

}

public static void brute(ArrayList arr) {

ArrayList distArr = getDistinct(arr);

```

```

List truthValues = truthTable(distArr.size());
int max = 0;
int index = -1;
for (int i = 0; i < truthValues.size(); i++) {
    String truth[] = truthValues.get(i).toString().split(",");
    int sat = Integer.parseInt(max2sat(arr, distArr, truth).toString());
    if (sat > max) {
        max = sat;
        index = i;
    }
}

```

```

String truth[] = truthValues.get(index).toString().split(",");
for (int i = 0; i < truth.length; i++) {
    if (truth[i].equals("1")) {
        System.out.print("T ");
    } else {
        System.out.print("F ");
    }
}
System.out.print(max);
System.out.println();
}

```

```

public static Integer max2sat(ArrayList arr, ArrayList distArr, String truth[]) {

```

```
ArrayList ref = new ArrayList();
```

```
for (int i = 0; i < distArr.size(); i++) {  
    int t = Integer.parseInt(truth[i]);  
    int d = Integer.parseInt(distArr.get(i).toString());  
    ref.add(t * d);  
}
```

```
int count = 0;  
for (int i = 0; i < arr.size(); i = i + 2) {  
    int x = Integer.parseInt(arr.get(i).toString());  
    int y = Integer.parseInt(arr.get(i + 1).toString());  
    boolean xFlag = true;  
    boolean yFlag = true;  
    for (int j = 0; j < ref.size(); j++) {  
        int r = Integer.parseInt(ref.get(j).toString());  
        if (Math.abs(x) == Math.abs(r)) {  
            x = x * r;  
        }  
        if (Math.abs(y) == Math.abs(r)) {  
            y = y * r;  
        }  
    }  
    if (x >= 0 || y >= 0) {  
        count++;  
    }  
}
```

```
return count;
```

```
}
```

```
private static List truthTable(int n) {
```

```
int rows = (int) Math.pow(2, n);
```

```
List arr = new ArrayList();
```

```
for (int i = 0; i < rows; i++) {
```

```
String ar = "";
```

```
for (int j = n - 1; j >= 0; j--) {
```

```
int x = (i / (int) Math.pow(2, j)) % 2;
```

```
if (x == 0) {
```

```
x = -1;
```

```
}
```

```
ar += x + ",";
```

```
}
```

```
arr.add(ar);
```

```
}
```

```
return arr;
```

```
}
```

```
public static ArrayList getDistinct(ArrayList arr) {
```

```
ArrayList distArr = new ArrayList();
```

```
for (int i = 0; i < arr.size(); i++) {
```

```
int x = Integer.parseInt(arr.get(i).toString());
```

```
boolean flag = true;
```

```
for (int j = 0; j < distArr.size(); j++) {
```

```
int y = Integer.parseInt(distArr.get(j).toString());
```

```
if (Math.abs(x) == Math.abs(y)) {
```

```
flag = false;
```

```
break;
```

```
}
```

```
}
```

```
if (flag) {
```

```
distArr.add(Math.abs(x));
```

```
}
```

```
}
```

```
return distArr;
```

```
}
```

```
public static void main(String h[]) throws Exception {
```

```
BufferedReader br = null;
```

```
try {
```

```
br = new BufferedReader(new FileReader("C:\\Users\\harishp647\\Desktop\\New  
folder\\test4.txt"));
```

```
String input = br.readLine();
```

```
if (input != null) {

    ArrayList arr = new ArrayList();
    while (input != null) {

        String str[] = input.split(" ");
        if(str.length==1){
            str = input.split("\t");
        }
        input = br.readLine();
        arr.add(str[0]);
        arr.add(str[1]);

    }
    boolean flag = true;

    Scanner reader = new Scanner(System.in);

    System.out.println();
    System.out.println("Press 1 for Brute-force approach");
    System.out.println("Press 2 for quick approach");
    System.out.println();
    System.out.print("Enter your selection: ");

    int choice = 4;
    try{
```

```
choice = reader.nextInt();
}catch(Exception e){
System.out.println("Invalid input");
}
System.out.println();
switch (choice) {

case 1:
brute(arr);
break;

case 2:
quick(arr);
break;

default:
System.out.println("Invalid input!!! You can only press 1, 2 or 3");
break;

}

} else {
System.out.println("The given file is empty!!!");
}
```



```
}catch(FileNotFoundException e){  
    System.out.println("Could not find the file specified!!");  
    System.out.println(e);  
}finally {  
    br.close();  
}  
}  
}
```