## **Cover Page**

Name: David Chen Salas

Section: 2023 Fall Term (1) Algorithms I CSCI 700 231[25504] (Queens College)

Project#: 6

Project Name: Partial ordering, dependency graph and Scheduling

**Due Date:** 11/17/2023 Friday before midnight

#### **Algorithm Steps:**

Step 1: inFile1, inFile2, outFile1, deBugFile open via args []

numNodes read from inFile1.

numProcs get from args [2]

if  $(numProcs \le 0)$ 

console write error message "need 1 or more processors".

exit the program

Else

if (numProcs > numNodes)

numProcs numNodes // means unlimited processors.

Step 2: Matrix dynamically allocate, size of numNodes+1 by numNodes+1

loadMatrix (inFile1, Matrix)

outFile1 "In main () after loadMatrix ()"

printMatrix (outFile1) // check for yourself to make sure the matrix is loaded correctly.

setMatrix (Matrix, deBugFile)

outFile1 "In main () after setMatrix ()"

printMatrix (outFile1) // check for yourself to make sure the matrix is correctly set.

Step 3: Open get a listNode (-99, 0, null) as dummy node for Open to point to

currentTime 0 // at the beginning of scheduling

procUsed 0 // no processor is used at the beginning

Step 4: totalJobTimes loadJobTimeAry (inFile2, jobTimeAry, deBugFile)

Table dynamically allocate, size of numProcs by totalJobTimes, initialize to zero.

outFile1 "in main() after allocate and initialize table"

printTable (outFile1, currentTime)

Step 5: scheduling (outFile1, deBugFile)

Step 6: outFile1 "in main() printing Table and Open"

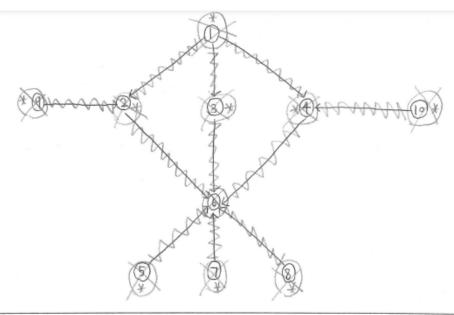
printTable (outFile1, currentTime) // The final schedule table.

printOpen (outFile1, currentTime)

Step 7: close all files

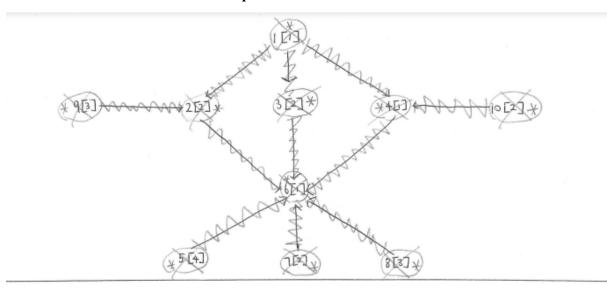
# **Illustration**

# \*\*Hand Draw Set1 with 3 processors\*\*



Time:	0	i	2	3	4	5	6	7	8	9	
Proc# 1:	١	8	3	6							
Proc#2:	5	9	2	~							
Proc#3:	7	10	4	-							

# \*\*Hand Draw Set2 with unlimited processors\*\*



Time:	0	1	2	3	4	5	6	7	8	9	
Proc#1:	1	3	3	2	2	6	San .				
Proc#2:	٦	٦	4	_	-						,
Proc#3:	0	10.	-	-	-	-					
Proc#4:	8	8	8	_	-	-					
Proc#4: Proc#5:	q	9	q	_	~	-					
Prac#6:	5	5	5	5	-	_					

$$\mathrm{open}\left(\mathrm{o}\right)\colon\left(1\mathcal{N}\right)\to\left(2\mathcal{N}\right$$

open (1): (3,2)

open (2): (AN)

open(3): (2)

open (4): -

open(5): (6,1)

open (6): empty

## **Source Code**

```
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.Scanner;
public class ChenSalasD_Project6_Main {
  static Scanner inFile1, inFile2;
  static int numProcessors;
  static FileWriter outFile1, deBugFile;
  static class listNode{
    int jobID;
    int jobTime;
    listNode next;
    listNode(int jobID, int jobTime, listNode next){
       this.jobID = jobID;
       this.jobTime = jobTime;
       this.next = next;
  static int numNodes;
  static int numProcs, procUsed;
  static int currentTime = 0;
  static int totalJobTimes;
  static int[] jobTimeAry;
  static int[][] Matrix;
  static int[][] Table;
  static listNode Open;
  public static void main(String[] args) throws IOException {
    inFile1 = new Scanner(new FileReader(args[0]));
     inFile2 = new Scanner(new FileReader(args[1]));
    numProcessors = Integer.parseInt(args[2]);
    outFile1 = new FileWriter(args[3]);
     deBugFile = new FileWriter(args[4]);
     numNodes = inFile1.nextInt();;
    numProcs = numProcessors;
    if(numProcs \le 0)
       System.out.println("Need 1 or more processors");
       System.exit(0);
     if(numProcs > numNodes){
       numProcs = numNodes;
    }
    //Step 2
     Matrix = new int[numNodes+1][numNodes+1];
    loadMatrix(inFile1, Matrix);
     outFile1.write("In main() after loadMatrix().\n");
    printMatrix(outFile1);
    setMatrix(Matrix, deBugFile);
    outFile1.write("\nIn main() after setMatrix().\n");
     printMatrix(outFile1);
```

```
//Step 3
  Open = new listNode(-99, 0, null);
  currentTime = 0;
  procUsed = 0;
  //Step 4
  jobTimeAry = new int[numNodes + 1];
  totalJobTimes = loadJobTimeAry(inFile2, jobTimeAry, deBugFile);
  Table = new int[numProcs + 1][totalJobTimes + 1];
  outFile1.write( "\nin main() after allocate and initialize table.\n");
  printTable(outFile1, currentTime);
  //Step 5
  scheduling(outFile1, deBugFile);
  //Step 6
  outFile1.write("In main() printing Table and Open\n");
  printTable(outFile1, currentTime);
  printOpen(outFile1, currentTime);
  inFile1.close();
  inFile2.close();
  outFile1.close();
  deBugFile.close();
public static void loadMatrix(Scanner inFile1, int[][] matrix) {
  while(inFile1.hasNext()){
     matrix[inFile1.nextInt()][inFile1.nextInt()] = 1;
public static void printMatrix(FileWriter outFile) throws IOException {
  for(int i = 0; i < numNodes + 1; i++){
     for (int j = 0; j < \text{numNodes} + 1; j++) {
            outFile.write(Matrix[i][j] + " ");
     outFile.write("\n");
public static void setMatrix(int[][] matrix, FileWriter deBugFile) throws IOException {
  deBugFile.write("Entering setMatrix method!\n");
  Matrix[0][0] = numNodes;
  for (int j = 1; j \le numNodes; j++) {
     matrix[0][j] = countParent(matrix, j, deBugFile);
     matrix[j][0] = countDependent(matrix, j, deBugFile);
  for (int i = 1; i \le numNodes; i++) {
     matrix[i][i] = 1;
  deBugFile.write("Before leaving setMatrix method, the Matrix below\n");
  printMatrix(deBugFile);
  deBugFile.write("Leaving serMatrix().\n");
public static int countParent(int[][] matrix, int j, FileWriter deBugFile) throws IOException {
  deBugFile.write("Entering countParent() method!\n");
  int count = 0;
  for(int i = 1; i \le numNodes; i++){
     if(matrix[i][j] > 0){
```

```
count++;
  deBugFile.write("In countParent() parent count is " + count);
  deBugFile.write("\nLeaving countParent() method.\n");
  return count;
public static int countDependent(int[][] matrix, int j, FileWriter deBugFile) throws IOException{
  deBugFile.write("Entering countDependent() method!\n");
  int count = 0;
  for(int i = 1; i \le numNodes; i++){
    if(matrix[j][i] > 0){
       count++;
  deBugFile.write("In countDependent() dependent count is " + count);
  deBugFile.write("\nLeaving countDependent() method.\n");
  return count;
public static int loadJobTimeAry(Scanner inFile, int[] jobTimeAry, FileWriter deBugFile) throws IOException {
  deBugFile.write("Entering loadJobTimeAry() method!\n");
  int jobTime = 0;
  int num = inFile.nextInt();
  for(int i = 1; i \le num; i++){
    jobTimeAry[inFile.nextInt()] = inFile.nextInt();
    jobTime += jobTimeAry[i];
    deBugFile.write(jobTime + "\n");
  deBugFile.write("Leaving loadJobTimeAry() method.\n");
  return jobTime;
public static void printTable(FileWriter outFile, int currentTime) throws IOException {
     for(int n = 0; n < 15+totalJobTimes*8; <math>n++) {
       outFile.write("=");
    outFile.write("\nProcUsed:" + procUsed + " currentTime:" + currentTime + "\n");
    outFile.write("Time:\t");
    for(int i = 0; i \le totalJobTimes; i++) {
       outFile.write(" "+i+"\t");
    outFile.write("\n");
    for(int n = 0; n < 15+totalJobTimes*8; n++) {
       outFile.write("-");
    outFile.write("\n");
     for(int i = 1; i \le numProcs; i++){
       outFile.write("Proc:" + i + "\t|");
       for(int j = 0; j \le currentTime; j++){
         outFile.write(" " + Table[i][j] + "\t|");
       outFile.write("\n");
       for(int n = 0; n < 15+totalJobTimes*8; n++) {
         outFile.write("-");
       outFile.write("\n");
```

```
printOpen(outFile, currentTime);
  }
  public static void printOpen(FileWriter outFile, int currentTime) throws IOException {
     listNode tmp = Open.next;
     outFile.write("Open(Time " + currentTime + "): ");
     while(tmp!=null){
       outFile.write("(" + tmp.jobID + ", " + tmp.jobTime + ")");
       if(tmp.next!=null){
         outFile.write(" --> ");
       tmp = tmp.next;
     outFile.write("\n");
  public static void scheduling(FileWriter outFile1, FileWriter deBugFile) throws IOException {
     deBugFile.write("Entering scheduling() method!\n");
     while(!isGraphEmpty()) {
       loadOpen();
       outFile1.write("In scheduling() after loadOpen currentTime is " + currentTime + "\n");
       printOpen(outFile1, currentTime);
       loadTable();
       outFile1.write("after loadTable currentTime is " + currentTime + "\n");
       printTable(outFile1, currentTime);
       currentTime++;
       printOpen(outFile1, currentTime);
       deleteDoneJobs(currentTime, deBugFile);
       if (checkCycle()) {
          outFile1.write("\n***There is cycle in the graph!!!***\n\n");
//
           System.exit(0);
          return;
     deBugFile.write("Leaving scheduling method.\n");
  public static void loadOpen() throws IOException {
     listNode spot;
     deBugFile.write("Entering loadOpen() method!\n");
     for(int j = 1; j \le numNodes; j++){
       if(Matrix[0][j] == 0 \&\& Matrix[j][j] == 1)
          Matrix[j][j] = 2;
          deBugFile.write("Find an orphan, the orphan is " + j + "\n");
          listNode newNode = new listNode(j, jobTimeAry[j], null);
          spot = findSpot(newNode);
          insertOneNode(spot, newNode);
     deBugFile.write("Before leaving loadOpen().\n");
     printOpen(deBugFile, currentTime);
     deBugFile.write("Leaving loadOpen() method.\n");
  public static listNode findSpot(listNode newNode) {
```

```
listNode spot = Open;
  while(spot.next != null && spot.next.jobTime < newNode.jobTime){</pre>
    spot = spot.next;
  return spot;
public static void insertOneNode(listNode spot, listNode newNode) {
  newNode.next = spot.next;
  spot.next = newNode;
public static listNode popNode(listNode open){
  listNode tmp = open.next;
  open.next = tmp.next;
  tmp.next = null;
  return tmp;
public static void loadTable() throws IOException{
  deBugFile.write("Entering loadTable() method!\n");
  printOpen(deBugFile, currentTime);
  int availProc = getNextProc(currentTime, deBugFile);
  while(availProc >= 0 && Open.next!=null && procUsed <= numProcs){
       listNode newNode = popNode(Open);
       putJobOnTable(availProc, currentTime, newNode.jobID, newNode.jobTime);
       if(availProc > procUsed){
         procUsed++;
       availProc = getNextProc(currentTime, deBugFile);
  deBugFile.write("Before leaving loadTable().\n");
  printTable(deBugFile, currentTime);
  printOpen(deBugFile, currentTime);
  deBugFile.write("Leaving loadTable() method.\n");
public static int getNextProc(int currentTime, FileWriter deBugFile) throws IOException {
  deBugFile.write("Entering getNextProc() method!\n");
  int i = 1;
  boolean notFound = true;
  while(notFound && i \le numNodes && i \le numProcs){
    if(Table[i][currentTime] == 0){
       deBugFile.write("in getNextProc() found an available proc#" + i + "\n");
       notFound = false;
       return i;
    i++;
  deBugFile.write("Leaving getNextProc() method. No available proc can be found.\n");
  return -1;
public static void putJobOnTable(int availProc, int currentTime, int jobID, int jobTime) {
  int Time = currentTime;
  int EndTime = Time + jobTime;
  while(Time < EndTime){
    Table[availProc][Time] = jobID;
    Time++;
}
public static void deleteDoneJobs(int currentTime, FileWriter deBugFile) throws IOException {
```

```
deBugFile.write("Entering deleteDoneJobs() method!\n");
  int proc = 1;
  while(proc <= procUsed) {</pre>
    if (Table[proc][currentTime] <= 0 && Table[proc][currentTime - 1] > 0) {
       int jobID = Table[proc][currentTime - 1];
       deleteOneJob(jobID, deBugFile);
    proc++;
  deBugFile.write("Leaving deleteDoneJobs() method below is the updated matrix.\n");
  printMatrix(deBugFile);
public static void deleteOneJob(int jobID, FileWriter deBugFile) throws IOException {
  deBugFile.write("Entering deleteOneJob() method!\n");
  Matrix[jobID][jobID] = 0;
  Matrix[0][0]--;
  for (int j = 1; j \le numNodes; j++) {
    if(Matrix[jobID][j] > 0){
       Matrix[0][j]--;
  deBugFile.write("Leaving deleteOneJob() method.\n");
public static boolean checkCycle() {
  if(Open.next == null && Matrix[0][0] > 0){
    for(int i = 1; i \le numProcs; i++){
       if(Table[i][currentTime-1] > 0){
         return false;
    return true;
  return false;
public static boolean isGraphEmpty() {
  if(Matrix[0][0] == 0){
    return true;
  return false;
```

# **Program Output**

# \*\*\*Set\_1 with 3 processors\*\*\*

```
**OutFile1**
In main() after loadMatrix().
0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0
In main() after setMatrix().
10 0 2 1 2 0 6 0 0 0 0
3 1 1 1 1 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 1 0 0 0
1 0 0 0 0 0 1 0 1 0 0
1 0 1 0 0 0 0 0 0 1 0
1 0 0 0 1 0 0 0 0 0 1
in main() after allocate and initialize table.
______
ProcUsed:0 currentTime:0
Time: 0 1 2 3 4 5 6 7 8 9 10
Proc:1 | 0 |
Proc:2 | 0 |
Proc:3 | 0 |
Open(Time 0):
In scheduling() after loadOpen currentTime is 0
Open (Time 0): (10, 1) --> (9, 1) --> (8, 1) --> (7, 1) --> (5, 1) --> (1, 1)
after loadTable currentTime is 0
______
ProcUsed:3 currentTime:0
Time: 0 1 2 3 4 5 6 7 8
                                                       9
Proc:1 | 10 |
Proc:2 | 9 |
Proc:3 | 8 |
______
Open(Time_0): (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
Open (Time 1): (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
In scheduling() after loadOpen currentTime is 1
```

```
Open (Time 1): (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
after loadTable currentTime is 1
______
ProcUsed:3 currentTime:1
                  3 4 5 6 7 8 9 10
Time: 0 1 2
Proc:1 | 10 | 7 |
Proc:2 | 9 | 5 |
Proc:3 | 8 | 1 |
Open(Time_1):
Open (Time 2):
In scheduling() after loadOpen currentTime is 2
Open (Time 2): (4, 1) \longrightarrow (3, 1) \longrightarrow (2, 1)
after loadTable currentTime is 2
ProcUsed: 3 currentTime: 2
Time: 0 1 2 3 4 5 6 7
Proc:1 | 10 | 7 | 4 |
Proc:2 | 9 | 5 | 3 |
______
Proc:3 | 8 | 1 | 2 |
Open (Time 2):
Open (Time 3):
In scheduling() after loadOpen currentTime is 3
Open (Time 3): (6, 1)
after loadTable currentTime is 3
______
ProcUsed:3 currentTime:3
                    3
                        4
                             5
                                       7
      0 1 2
                                  6
                                           8
Time:
Proc:1 | 10 | 7 | 4 | 6 |
Proc:2 | 9 | 5 | 3 | 0 |
Proc:3 | 8 | 1 | 2 | 0 |
Open(Time_3):
Open (Time 4):
In main() printing Table and Open
ProcUsed:3 currentTime:4
Time: 0 1 2 3 4 5 6 7 8 9 10
Proc:1 | 10 | 7 | 4 | 6 | 0 |
Proc:2 | 9 | 5 | 3 | 0 | 0 |
```

#### \*\*deBugFile\*\*

Entering setMatrix method! Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 3 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 2 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 2 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 6 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 0 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method!

```
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Before leaving setMatrix method, the Matrix below
10 0 2 1 2 0 6 0 0 0 0
3 1 1 1 1 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 1 1 0 0 0
1 0 0 0 0 0 1 0 1 0 0
1 0 1 0 0 0 0 0 0 1 0
1 0 0 0 1 0 0 0 0 0 1
Leaving serMatrix().
Entering loadJobTimeAry() method!
3
4
5
6
7
8
9
10
Leaving loadJobTimeAry() method.
Entering scheduling() method!
Entering loadOpen() method!
Find an orphan, the orphan is 1
Find an orphan, the orphan is 5
Find an orphan, the orphan is 7 Find an orphan, the orphan is 8
Find an orphan, the orphan is 9
Find an orphan, the orphan is 10
Before leaving loadOpen().
Open(Time_0): (10, 1) --> (9, 1) --> (8, 1) --> (7, 1) --> (5, 1) --> (1, 1)
Leaving loadOpen() method.
Entering loadTable() method!
Open (Time 0): (10, 1) \longrightarrow (9, 1) \longrightarrow (8, 1) \longrightarrow (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
ProcUsed:3 currentTime:0
Time: 0 1 2 3 4 5 6 7 8
```

Proc:1 | 10 |

```
Proc:2 | 9 |
______
Proc:3 | 8 |
Open(Time_0): (7, 1) --> (5, 1) --> (1, 1)
Open (Time 0): (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
7 0 1 1 1 0 5 0 0 0 0
3 2 1 1 1 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 2 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 2 0 0 0
1 0 0 0 0 0 1 0 0 0 0
1 0 1 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0 0
Entering loadOpen() method!
Before leaving loadOpen().
Open (Time 1): (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
Leaving loadOpen() method.
Entering loadTable() method!
Open(Time_1): (7, 1) \longrightarrow (5, 1) \longrightarrow (1, 1)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
=======
ProcUsed:3 currentTime:1
Time: 0 1 2 3 4 5 6 7 8
                                                          9
                                                                10
Proc:1 | 10 | 7 |
Proc:2 | 9 | 5 |
______
Proc:3 | 8 | 1 |
                  _____
Open (Time 1):
Open (Time 1):
Leaving loadTable() method.
```

```
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
4 0 0 0 0 0 3 0 0 0 0
3 0 1 1 1 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 1 0 0 0 0
1 0 1 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0
Entering loadOpen() method!
Find an orphan, the orphan is 2
Find an orphan, the orphan is 3
Find an orphan, the orphan is 4
Before leaving loadOpen().
Open(Time_2): (4, 1) --> (3, 1) --> (2, 1)
Leaving loadOpen() method.
Entering loadTable() method!
Open(Time_2): (4, 1) --> (3, 1) --> (2, 1)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
ProcUsed:3 currentTime:2
Time: 0 1 2
                             4 5
                        3
                                         6 7
                                                     8
                                                           9
Proc:1 | 10 | 7 | 4 |
______
Proc:2 | 9 | 5 | 3 |
                          ._____
Proc:3 | 8 | 1 | 2 |
Open (Time 2):
Open (Time 2):
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
```

```
1 0 0 0 0 0 0 0 0 0
3 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 0 0 0
1 0 0 0 0 0 1 0 0 0 0
1 0 1 0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0 0 0
Entering loadOpen() method!
Find an orphan, the orphan is 6
Before leaving loadOpen().
Open(Time_3): (6, 1)
Leaving loadOpen() method.
Entering loadTable() method!
Open (Time 3): (6, 1)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Before leaving loadTable().
______
ProcUsed:3 currentTime:3
Time: 0 1 2
                                         4
                                                5
                                                      6 7 8 9
                                                                                     10
***Set 2 with unlimited processors***
**OutFile1**
In main() after setMatrix().
10 0 2 1 2 0 6 0 0 0 0
ProcUsed:0 currentTime:0
Time: 0
12
                                                                                    10
Proc:2
Proc:4
Proc:5
Proc:6
Proc:7
Open(Time 0):
In scheduling() after loadOpen currentTime is 0
Open(Time 0): (1, 1) -> (10, 2) --> (7, 2) --> (9, 3) --> (8, 3) --> (5, 4)
after loadTable currentTime is 0
Proc:1
Proc:2
Proc:3
Proc:5
```

roc:6  roc:7	I 5										
	1 0										
:oc:8		<u>I</u>									
:oc:9	I 0										
roc:10	I 0										
en (Time 1)	: ug() after loadOpe		s 1								
ime:	currentTime:1	1	2	3	4	5	6	7	8	9	10
	13	14	2 15	3 16	4 17	5 18	6 19	7 20	21		
oc:1	1	3	ı								
oc:2	10	10	I								
c:3	7	1 7									
c:4	9	9									
c:5	8	8									
oc:6	5	1 5									
c:7	I 0	1 0	I								
oc:8	1 0	1 0									
oc:9	1 0	1 0									
c:10	1 0	1 0	I								
en(Time_2) ter loadTa	: ug() after loadOpe		s 2								
me:	0	1 14	2 15	3 16	4 17	5 18	6 19	7 20	8 21	9	10
oc:1	1	3	3								
oc:2			4	I							
	10	10									
c:3	7	7	I 0								
c:4	J 9	l 9	l 9								
c:5	8	8	8								
c:6	5	5	5	ı							
c:7	1 0	1 0	1 0								
oc:8	1 0	1 0	1 0								
c:9	1 0	1 0	0	I							
en (Time_2) en (Time_3) schedulir en (Time 3)	: ng() after loadOpe : (2, 2)		s 3	I							
en (Time_2) en (Time_3) schedulir en (Time_3) eer loadTa	: : :g() after loadOpe	en currentTime i		3 16	4 17	5 18	6 19	7 20	8 21	9	10
en (Time_2) en (Time_3) schedulir en (Time_3) eer loadTa	: :: :g() after loadOpe : (2, 2) :ble currentTime: currentTime:3	en currentTime i	s 3 2	3	4 17	5 18	6 19	7 20	8 21	9	10
n (Time_2) n (Time_3) n (Time_3) er loadTa cUsed:6	: : : : : : : : : : : : : : : : : : :	en currentTime i is 3	2 15	3 16	17	5 18	6 19	7 20	8 21	9	10
n(Time_2) n(Time_3) schedulir n(Time_3) er loadTa cUsed:6 e:	gg() after loadOpp : (2, 2) ble currentTime: currentTime:3 0 13	en currentTime i is 3  1 14   3	2 15	3 16   2	17	5 18	6 19	7 20	8 21	9	10
n (Time_2) n (Time_3) schedulir n (Time_3) er loadTe cUsed:6 e:	: : : : : : : : : : : : : : : : : : :	1 14 1 3 1 10 1 7	2 15 1 3 1 4	3 16 1 2 1 0	17   	5 18	6 19	7 20	8 21	9	10
n (Time_2) n (Time_3) schedulir n (Time_3) schedulir n (Time_3) er loadTs cUsed:6 e: c:1 c:2 c:3 c:4	:: :g() after loadOpe : (2, 2) bble currentTime: 0 13   1   1   7	en currentTime i 1s 3  1 14  1 3  1 10  1 7	2 15 1 3 1 4 1 0	3 16 1 2 1 0	17   	5 18	6 19	7 20	8 21	9	10
en (Time 2) en (Time 3) schedulir en (Time 3) schedulir en (Time 3) cused:6 ec:	:: g() after loadOp(: (2, 2) ble currentTime: CurrentTime:3 0 13   1 10 17	1 14 1 3 1 10 1 7 1 9 1 8	2 15 1 3 1 4 1 0	3 16 1 2 1 0	17	5 18	6 19	7 20	8 21	9	10
n (Time_2) n (Time_3) schedulir schedulir n (Time_3) er loadTe ccUsed:6 e: c:1 c:2 c:3 c:4	:: :g() after loadOpe : (2, 2) bble currentTime: 0 13   1   1   7	en currentTime i 1s 3  1 14  1 3  1 10  1 7	2 15 1 3 1 4 1 0	3 16 1 2 1 0	17   	5 18	6 19	7 20	8 21	9	10
n (Time 2) n (Time 3) schedulir schedulir n (Time 3) schedulir n (Time 3) schedulir n (Time 3) schedulir cutsed:6 e: c:1 c:2 c:3 c:4 c:5	:: g() after loadOp(: (2, 2) ble currentTime: CurrentTime:3 0 13   1 10 17	1 14 1 3 1 10 1 7 1 9 1 8	2 15 1 3 1 4 1 0	3 16 1 2 1 0	17	5 18	6 19	7 20	9 21	9	10
en (Time 2) en (Time 3) en (Time 4) en (Time 2) en (Time 4) en (Time 2) en (Time 3) en (Time 4) en (Time 2) en (Time 2) en (Time 2) en (Time 4) en (Ti	: ; ; ; () after loadOp (2, 2) ble currentTime : 3	1 14 1 3 1 10 1 7 1 9 1 8 1 5	2 15 1 3 1 4 1 0 1 9	3 16 1 2 1 0 1 0 1 0	17   	5 18	6 19	7 20	8 21	9	10
en (Time 2) en (Time 3) en (Time 3) schedulin en (Time 3) ter loadre cocused:6 me: coc:1 coc:2 coc:3 coc:4 coc:6 coc:7 coc:8	: : : : : : : : : : : : : : : : : : :	1 14 1 10 1 10 1 10 1 10 1 10 1 10 1 10	2 15 1 3 1 4 1 0 1 9 1 8 1 5	3 16 1 2 1 0 1 0 1 0 1 0 1 5 1 0 0 1 5 1 0 0 1 0 1	17 	5 18	6 19	7 20	8 21	9	10
en (Time 2) en (Time 3) en (Time 3) en (Time 3) schedulir en (Time 3) ter loadTe ocUsed:6 me: oc:1 oc:2 oc:3 oc:4 oc:5 oc:6 oc:7 oc:8	: : : : : : : : : : : : : : : : : : :	1 14 1 3 1 10 1 1 9 1 8 1 5 1 0 0 1 0 0 1 0 0 1 0 0	2 15 1 3 1 4 1 0 1 9 1 8 1 5 1 0 1 0 1 0 0 1 0	3 16 1 2 1 0 1 0 1 0 1 5 1 0 1 0 1 0 1 0 1 0 1 0	17	5 18	6 19	7 20	8 21	9	10
pen (Time 3) tfer loadTe roccUsed:6 ime: 2 rocc:1 rocc:2 rocc:3 rocc:4 rocc:5 rocc:6 rocc:7 rocc:8 rocc:9 rocc:10 pen (Time 3) roccitume 3) roccitume 3)	: : g() after loadOp((2, 2) ble currentTime: 3	1 14 1 3 1 10 1 7 1 8 1 1 5 1 0 1 0 1 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0	2 15 1 3 1 4 1 0 1 9 1 8 1 5 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 1 0 1 0 1 5 1 0 1 0 1 0 1 0 1 0 1 0	17	5 18	6 19	7 20	8 21	9	10
en(Time 2) en(Time 3) en(Time 3) schedulir en(Time 3) ene ene ene ene ene ene ene ene ene en	: ; ; () after loadOp((2, 2) ble currentTime: 3	1 14 1 10 1 7 1 9 1 8 1 5 1 0 1 0 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1	2 15 1 3 1 4 1 0 1 9 1 8 1 5 1 0 1 0 1 0 0 1 0 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18	19	20	21	9	10
en (Time 2) en (Time 3) en (Time 4)	: : : : : : : : : : : : : : : : : : :	en currentTime i 1s 3  1 14  1 3  1 10  1 7  1 9  1 8  1 5  1 0  1 0  1 0  1 0  1 10	2 15 15 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	3 16 1 2 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18	19	7 20	21		
in (Time_2) in (Time_2) in (Time_3) in (Time_4) in (Ti	: : : : : : : : : : : : : : : : : : :	en currentTime i 1s 3  1 14  1 3  1 10  1 7  1 9  1 8  1 5  1 0  1 0  1 0  1 0  1 1 0  1 1 0  1 1 0  1 1 0	2 15 1 3 1 4 1 0 0 1 9 1 8 1 5 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18 	19	20	21		
n (Time 2) n (Time 2) n (Time 2) n (Time 3) a cachedul ir (Time 3) n (Time 4)	: : : : : : : : : : : : : : : : : : :	1 14 1 3 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 1 0 0	2 15 1 3 1 4 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 5 1 5	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18	19	20	21		
(Time 2) (Time 3) (Time 2) (Time 3) (Time 4) (Ti	: : : : : : : : : : : : : : : : : : :	en currentTime i  ta 3  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 15 1 3 1 4 1 0 0 1 9 1 8 1 5 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 0 1 0 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18 	19	20	21		
(Time 2) (Time 3) (Time 4) (Ti	: : : : : : : : : : : : : : : : : : :	1 14 1 3 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 1 0 0	2 15 1 3 1 4 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 5 1 5	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18	19	20	21		
(Time 2) (Time 3) (Time 2) (Time 3) (Time 4) (Ti	:: ;; g() after loadOp: (2, 2)   ; (2, 2)   ; (2, 2)   ; (2, 2)   ; (2, 2)   ; (3, 2)   ; (3, 2)   ; (3, 2)   ; (4, 2)   ; (5, 2)   ; (7, 2)   ; (8, 2)   ; (9, 2)   ; (1, 2)   ; (1, 2)   ; (1, 2)   ; (1, 2)   ; (1, 2)   ; (2, 2)   ; (3, 2)   ; (4, 2)   ; (5, 2)   ; (7, 2)   ; (8, 2)   ; (1, 2)	en currentTime i is 3  1 14  1 3  1 10  1 7  1 9  1 8  1 5  1 0  1 0  1 0  1 0  1 1 0  1 1 0  1 1 0  1 1 0	2 15 1 3 1 4 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18 5 18	19	20	21		
(Time 2) (Time 3) (Time 2) (Time 3) (Time 4) (Time 3) (Time 4) (Time 3) (Time 4) (Ti	:: ;; g() after loadOp: ; (2, 2) ; g() after loadOp: ; (2, 2) ; (2, 2) ; (3, 2)	en currentTime i is 3  1 14  1 3  1 10  1 7  1 9  1 8  1 5  1 0  1 0  1 0  1 1 0  1 1 0  1 1 0  1 1 0  1 1 0	2 15 1 3 1 4 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18	19	20	21		
(Time_2)	: : : : : : : : : : : : : : : : : : :	en currentTime i ts 3  1 1 1 1 1 1 1 1 7 1 9 1 8 1 5 1 0 1 0 1 0 1 0 1 1 0 1 1 7 1 1 1 1 1 1	2 15 1 3 1 4 1 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 1 0 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17	18 	19	20	21		
in (Time_2) in (Time_2) in (Time_2) in (Time_3) in (Ti	: ; ; () after loadOp (2, 2) ble currentTime: 3	1 14 1 3 1 10 1 10 0 1 10 0 1 1 14 1 1 3 1 1 10 1 1 1 1 1 1 1 1 1 1 1 1 1	2 15 1 0 1 0 0 1 0 0 1 9 1 8 1 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 9 1 8 1 5 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 1 0 0 1 1 9 1 1 8 1 1 5 1 1 1 0 0 1 1 9 1 1 8 1 1 1 5 1 1 1 0 0 1 1 1 9 1 1 1 1 1 1 1 1 1 1 1	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1 1	19	20	21		
en (Time 2) en (Time 3) en (Ti	:: ;; () after loadOp ((2, 2) ble currentTime: 3	en currentTime i 1s 3  1 14  1 3  1 9  1 8  1 0  1 0  1 0  1 0  1 1 0	2 15 1 0 1 0 3 4 4 1 0 0 1 9 1 8 4 1 5 1 5 1 0 1 9 1 8 1 5 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1	19	20	21		
en (Time_2) en (Time_2) en (Time_3) en (Ti	:: ;; () after loadOp ((2,2) ble currentTime: 3	en currentTime i 1s 3  1 14  1 3  1 10  1 7  1 9  1 8  1 5  1 0  1 0  1 7  1 0  1 1 0  1 7  1 1 9  1 8  1 1 5  1 0  1 0  1 0  1 0  1 0  1 0  1	2 15 1 0 1 0 1 0 1 0 1 9 1 8 4 1 5 1 5 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1	19	20	21		
en (Time_2) en (Time_2) en (Time_3) en (Ti	:: ;; () after loadOp ((2, 2) ble currentTime: 3	en currentTime i 1s 3  1 14  1 3  1 9  1 8  1 0  1 0  1 0  1 0  1 1 0	2 15 1 0 1 0 3 4 4 1 0 0 1 9 1 8 4 1 5 1 5 1 0 1 9 1 8 1 5 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1	19	20	21		
pen (Time_2) pen (Time_3) pen (Time_3) a schedulir a schedulir tter loadTe	: ; ; () after loadOp ((2, 2) ble currentTime : (2, 2) ble currentTime : 3	en currentTime i  ts 3  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 15 1 3 1 4 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1	19	20	21		
in (Time 2) in (Time 2) in (Time 3) in (Time 4) in (Ti	: ; ; () after loadOp (2, 2) ble currentTime : (2, 2) bble currentTime : (2, 2) bble currentTime : (3, 0)	1 14 1 10 1 7 1 9 1 8 1 10 1 10 1 7 1 9 1 8 1 1 10 1 1 7 1 9 1 8 1 1 10 1 1 7 1 1 9 1 1 8 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 15 1 0 1 0 0 1 0 0 1 9 1 8 1 1 5 1 0 0 1 9 1 8 1 1 5 1 1 0 0 1 9 1 8 1 1 5 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1 1 1 1 1	6 19	7 20	21 8 21	9	10
en (Time 2) en (Time 3) en (Time 4) en (Time 5) en (Time 5) en (Time 6) en (Ti	: ; ; () after loadOp ((2, 2) ble currentTime : (2, 2) ble currentTime : 3	en currentTime i  1s 3  1 14  1 3  1 10  1 7  1 9  1 8  1 5  1 0  1 0  1 0  1 7  1 9  1 8  1 15  1 0  1 0  1 0  1 0  1 0  1 0  1	2 15 1 3 1 4 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1	19	7 20	21		
en (Time_2) en (Time_3) en (Time_3) en (Time_3) en (Time_3) en (Time_3) en (Time_3) en (Time_4) en (Time_4) en (Time_4) en (Time_4) en (Time_4) en (Time_5) en (Time_4) en (Time_5) en (Time_4) en (Time_5) en (Time_4) en (Time_5) en (Time_4)	: ; ; () after loadOp (2, 2) ble currentTime : (2, 2) bble currentTime : (2, 2) bble currentTime : (3, 0)	1 14 1 10 1 7 1 9 1 8 1 10 1 10 1 7 1 9 1 8 1 1 10 1 1 7 1 9 1 8 1 1 10 1 1 7 1 1 9 1 1 8 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2 15 1 0 1 0 0 1 0 0 1 9 1 8 1 1 5 1 0 0 1 9 1 8 1 1 5 1 1 0 0 1 9 1 8 1 1 5 1 1 0 0 1 1 0 0 1 1 0 0 0 1 0 0 0 1 0 0 0 1 0	3 16 1 2 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	17  1  1  1  1  1  1  1  1  1  1  1  1	18 5 18 1 1 1 1 1	6 19	7 20	21 8 21	9	10

Proc:3	1 7	7	0	1 0	1 0	1 0					
Proc:4			y	I 0							
Proc:5	8	8	8	1 0	1 0	1 0	1				
Proc:6		1 5	1 5	1 5							
Proc:7	1 0	0	1 0	0	1 0	0	1				
Proc:8	1 0	1 0	1 0	1 0	1 0	1 0					
Proc:9	1 0	1 0	1 0	1 0	1 0	1 0					
Proc:10	1 0	0	1 0	1 0	0	0	1				
Open(Time_5): Open(Time_6): In main() pri	nting Table and	Open									
ProcUsed:6 c	urrentTime:6										
Time:	0	1	2	3	4	5	6	7	8	9	10
12	13	14	15	16	17	18	19	20	21		
Proc:1	1	1 3	1 3	1 2	2	6	1 0	I			
Proc:2	10	10	4	0	1 0	1 0	1 0	ı			
Proc:3	1 7	1 7	1 0	1 0	1 0	1 0	1 0	I			
Proc:4	J 9	l 9	l 9	1 0	1 0	1 0	1 0	I			
Proc:5	8	8	8	1 0	1 0	1 0	I 0				
Proc:6	J 5	ı 5	ı 5	, 5	I 0	I 0	I 0	I			
Proc:7		1 0	1 0	1 0		1 0					
Proc:9		I 0	I 0	I 0	1 0	I 0	I 0				
Proc:10	1 0	1 0	1 0	1 0	1 0	1 0	1 0				
Open (Time_6): Open (Time_6):											

#### \*\*deBugFile\*\*

Entering setMatrix method! Entering countParent() method! In countParent() parent count is 0Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 3 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 2 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 2 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 6 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 0 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0

```
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Before leaving setMatrix method, the Matrix below
10 0 2 1 2 0 6 0 0 0 0
3 1 1 1 1 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 1 0 0 0
1 0 0 0 0 0 1 0 1 0 0
1 0 1 0 0 0 0 0 0 1 0
1 0 0 0 1 0 0 0 0 0 1
Leaving serMatrix().
Entering loadJobTimeAry() method!
3
5
6
10
11
13
16
19
21
Leaving loadJobTimeAry() method.
Entering scheduling() method!
Entering loadOpen() method!
Find an orphan, the orphan is 1
Find an orphan, the orphan is 5
Find an orphan, the orphan is 7
Find an orphan, the orphan is 8
Find an orphan, the orphan is 9
Find an orphan, the orphan is 10
Before leaving loadOpen().
Open(Time_0): (1, 1)^- -> (10, 2) --> (7, 2) --> (9, 3) --> (8, 3) --> (5, 4)
Leaving loadOpen() method.
Entering loadTable() method!
Open (Time 0): (1, 1) \longrightarrow (10, 2) \longrightarrow (7, 2) \longrightarrow (9, 3) \longrightarrow (8, 3) \longrightarrow (5, 4)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
```

```
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
in getNextProc() found an available proc#4
Entering getNextProc() method!
in getNextProc() found an available proc#5
Entering getNextProc() method!
in getNextProc() found an available proc#6
Entering getNextProc() method!
in getNextProc() found an available proc#7
Before leaving loadTable().
______
______
ProcUsed:6 currentTime:0
Time: 0 1 2
            2 3 4 5 6 7 8
15 16 17 18 19 20 21
                                                 11
        14
     13
______
Proc:1 | 1 |
______
Proc:4 | 9 |
Proc:5 | 8 |
Proc:6 | 5 |
Proc: 7 | 0 |
______
Proc:8 | 0 |
Proc:9 | 0 |
Proc:10 | 0 |
Open (Time 0):
Open(Time_0):
```

in getNextProc() found an available proc#2

```
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
9 0 1 0 1 0 6 0 0 0 0
3 0 1 1 1 0 0 0 0 0 0
1 0 1 0 0 0 1 0 0 0 0
1 0 0 1 0 0 1 0 0 0 0
1 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 2 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0
1 0 0 0 0 0 1 2 0 0 0
1 0 0 0 0 0 1 0 2 0 0
1 0 1 0 0 0 0 0 0 2 0
1 0 0 0 1 0 0 0 0 0 2
Entering loadOpen() method!
Find an orphan, the orphan is 3
Before leaving loadOpen().
Open(Time 1): (3, 2)
Leaving loadOpen() method.
Entering loadTable() method!
Open(Time_1): (3, 2)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#7
Before leaving loadTable().
______
______
                                J 6
18 11
ProcUsed:6 currentTime:1
                     3 4 5
16 17 18
                                               8
Time: 0 1 2
                                          7
                                                          10
                                                               11
               15
                                     19
       13
           14
                                          20
                                                21
Proc:1 | 1 | 3 |
Proc:2 | 10 | 10 |
Proc:3 | 7 | 7 |
______
Proc:4 | 9 | 9 |
Proc:5 | 8 | 8 |
Proc:6 | 5 | 5 |
```

# \*\*\*Set\_3 with 3 processors\*\*\*

#### \*\*OutFile1\*\*

In main() after loadMatrix().

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 1
In main() after setMatrix().
14 0 1 1 1 1 5 5 4 1 0 0 0 0 2
3 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0
2 0 0 0 0 0 0 1 1 0 0
1 0 0 0 0 0 0 0 1 0 0
                 1 0 0 0
                 0 1 0 0
1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1
in main() after allocate and initialize table.
ProcUsed:0 currentTime:0
     0 1
13 14
Time:
                                                                                               11
12
                         15
                                 16
                                         17
                                                18
                                                        19
                                                                        21
                                                                                22
                                                                                               24
25
          26
                 27
Proc:1 | 0 |
Proc:2 | 0 |
Proc:2
Open(Time_0):
In scheduling() after loadOpen currentTime is 0
Open(Time_0): (12, 1) --> (10, 1) --> (11, 2) --> (1, 2) --> (13, 4) after loadTable currentTime is 0
ProcUsed:3 currentTime:0
      0 1
13 14
                         2
15
                                 3
16
                                                5
18
                                                        6
19
                                                                        8
21
Time:
                                                                               9
22
                                                                                               11
                                                                20
12
                                                                                               24
25
----
Proc:1
      | 12
Proc:2
       1 10
Open(Time_0): (1, 2) --> (13, 4)
Open(Time_1): (1, 2) --> (13, 4)
In scheduling() after loadOpen currentTime is 1
Open(Time_1): (1, 2) --> (13, 4) after loadTable currentTime is 1
ProcUsed:3 currentTime:1
Time:
     0 1
13 14
                                                                                               11
12
                 14
                                 16
                                         17
                                                 18
                                                        19
25
          26
                 27
_____
Proc:1 | 12
               | 1
                       1
               | 13
Proc:2 | 10
          ----
Proc:3 | 11
               | 11
                     - 1
Open(Time_1):
```

Open (Time 2): In scheduling() after loadOpen currentTime is 2 Open (Time 2): after loadTable currentTime is 2 ProcUsed:3 currentTime:2 Time: 0 1 12 13 14 15 16 17 18 19 14 27 25 26 Proc:1 | 12 | 1 Proc:2 | 10 | 13 | 13 Open (Time 2): Open (Time\_3): Open(Time\_3): (3, 1) --> (4, 2) --> (2, 3) after loadTable currentTime is 3 ProcUsed:3 currentTime:3 Time: 0 1 2
12 13 14 15
25 26 27 9 10 22 23 16 17 18 19 20 21 24 | 13 1 13 Proc:3 | 11 | 11 | 0 | 4 \_\_\_\_\_\_ Open(Time\_3): (2, 3)
Open(Time\_4): (2, 3)
In scheduling() after loadOpen currentTime is 4 Open(Time\_4): (2, 3) after loadTable currentTime is 4 ProcUsed:3 currentTime:4 Time: 0 1 2 3 4 5 6 12 13 14 15 16 17 18 19 25 26 27 20 21 22 24 Proc:1 | 12 | 1 | 1 | 3 | 2 Proc:2 | 10 | 13 | 13 | 13 | 13 | ... Open(Time\_4): Open (Time 5): In scheduling() after loadOpen currentTime is 5 Open (Time 5): after loadTable currentTime is 5 -----ProcUsed:3 currentTime:5 Time: 0 1 12 13 14 25 26 27 11 15 16 17 18 19 20 Proc;2 | 10 | 13 | 13 | 13 | 0 | Proc:3 | 11 | 11 | 0 | 4 | 4 | 0 Open (Time 5): Open (Time\_6): In scheduling() after loadOpen currentTime is 6 Open (Time 6): after loadTable currentTime is 6

Time: 12 25		0 13 26		1 14 27						4 17		5 18		6 19		7 20	8 21		9 22	10 23	11 24
Proc:1	 	12	 	1		1		3		2		2	   		 						
Proc:2										13			   		 				- 		
Proc:3	1	11	1	11	1	0	1	4	1	4	- 1	0	1	0	- 1				=		
	dulin ne_7) padTa ===== d:3	g() af : ble cu	ırren	tTime	is 7	2	.====	3		4		5					8	=====		10 23	11 24
Proc:1		12	 	1		1		3	1	2	ı	2	 	2	 	0	 				
		10	 	13	I	13	I	13	l	13		0	 	0	 	0	l		_		
Proc:3						0		4				0		 0	 				-		
		11		11		-		-		-			'								

#### \*\*\*There is cycle in the graph!!!\*\*\*

ProcUsed:3 Time: 12		current 0 13		1		2 15		3		4		5 18		6 19		7		8 21	9 22	10 23	11 24
12 25 		26 		27		15		10		1/		10		19						 	
Proc:1	 	12		1		1		3	 	2	   	2		2		0		0	   	 	
Proc:2		10		13	 	13	   	13		13	   	0	 	0	 	0	   	0	   	 	
Proc:3	 	11	 	11	 I	0	I	4		4		0		0		0	 I	0	 		

#### \*\*deBugFile\*\*

Entering setMatrix method! Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 3 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2

Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 5 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 5 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 4 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 2

```
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Before leaving setMatrix method, the Matrix below
14 0 1 1 1 1 5 5 4 1 0 0 0 0 2
3 1 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 0 0 0
2 0 0 0 0 0 1 0 1 0 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 0 0 1 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0
2 0 0 0 0 0 1 0 0 0 1 0 0 0 1
2 0 0 0 0 0 0 1 1 0 0 1 0 0 0
1 0 0 0 0 0 0 0 1 0 0 0 1 0 0
1 0 0 0 0 1 0 0 0 0 0 1 0
1 0 0 0 0 1 0 0 0 0 0 0 0 1
Leaving serMatrix().
Entering loadJobTimeAry() method!
5
6
8
9
11
12
14
17
18
20
21
25
27
Leaving loadJobTimeAry() method.
Entering scheduling() method!
Entering loadOpen() method!
Find an orphan, the orphan is 1
Find an orphan, the orphan is 10
Find an orphan, the orphan is 11
Find an orphan, the orphan is 12
Find an orphan, the orphan is 13
Before leaving loadOpen().
Open (Time 0): (12, 1) \longrightarrow (10, 1) \longrightarrow (11, 2) \longrightarrow (1, 2) \longrightarrow (13, 4)
Leaving loadOpen() method.
Entering loadTable() method!
Open(Time 0): (12, 1) --> (10, 1) --> (11, 2) --> (1, 2) --> (13, 4)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
```

\_\_\_\_\_

```
2
                  3
                       4
                           5
                               6 7
                                                 10 11
Time:
12
      13
         14
              15
                   16
                       17
                           18
                                19
                                    20
                                         21
2.5
      26
          2.7
______
______
Proc:1 | 12 |
______
_____
Proc:2 | 10 |
Proc:3 | 11 |
______
______
_____
Open (Time 0): (1, 2) \longrightarrow (13, 4)
Open (Time 0): (1, 2) \longrightarrow (13, 4)
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
12 0 1 1 1 1 4 5 3 1 0 0 0 0 1
3 2 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 0 0 0
2 0 0 0 0 1 0 1 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 0 0 0 0 0
1 0 0 0 0 0 0 1 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0
2000010000001
2 0 0 0 0 0 0 1 1 0 0 2 0 0 0
1 0 0 0 0 0 0 0 1 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 0 2 0
1 0 0 0 0 1 0 0 0 0 0 0 0 1
Entering loadOpen() method!
Before leaving loadOpen().
Open (Time 1): (1, 2) \longrightarrow (13, 4)
Leaving loadOpen() method.
Entering loadTable() method!
Open (Time_1): (1, 2) \longrightarrow (13, 4)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
_____
```

ProcUsed:3 currentTime:1

#### \*\*\*Set\_4 with 3 processors\*\*\*

#### \*\*OutFile1\*\*

In main() after loadMatrix().

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 1
In main() after setMatrix().
14 0 1 1 1 0 5 4 4 1 0 0 0 0 2
0 0 0
2 0 0 0 0 0 0 1 1 0 0
1 0 0 0 0 0 0 0 1 0 0
              1 0 0 0
              0 1 0 0
in main() after allocate and initialize table.
ProcUsed:0 currentTime:0
       0
13
             1
14
Time:
                                                                                 11
12
                     15
                            16
                                  17
                                         18
                                               19
                                                             21
                                                                   22
                                                                          23
                                                                                 24
25
        26
               27
                     28
Proc:1 | 0 |
Proc:2 | 0 |
Proc:2
Open(Time_0):
In scheduling() after loadOpen currentTime is 0
Open(Time_0): (13, 1) --> (10, 1) --> (5, 1) --> (12, 2) --> (1, 2) --> (11, 4) after loadTable currentTime is 0
ProcUsed:3 currentTime:0
     0 1
13 14
                     2
                           3
16
                                         5
18
                                               6
19
                                                             8
21
                                                                   9
22
Time:
                                  4
17
                                                                                 11
                     15
                                                      20
12
                                                                                 24
25
                     28
----
               -----
Proc:1
     | 13
Proc:2
      1 10
Open(Time_0): (12, 2) --> (1, 2) --> (11, 4)
Open(Time_1): (12, 2) --> (1, 2) --> (11, 4)
In scheduling() after loadOpen currentTime is 1
Open(Time_1): (12, 2) --> (1, 2) --> (11, 4) after loadTable currentTime is 1
ProcUsed:3 currentTime:1
Time:
       0
13
                                                                                 11
12
               14
                     15
                            16
                                  17
                                         18
                                                19
                                                                   22
25
        26
               27
                     28
______
Proc:1 | 13
             | 12
                   1
Proc:2 | 10
             | 1
Proc:3 | 5
             | 11
                  - 1
Open(Time_1):
```

Open (Time 2): In scheduling() after loadOpen currentTime is 2 Open (Time 2): after loadTable currentTime is 2 ProcUsed:3 currentTime:2 Time: 0 1 12 13 14 14 15 16 17 27 28 18 19 22 25 26 Proc:1 | 13 | 12 | 12 | Proc:2 | 10 Open (Time 2): Open (Time\_3): Open(Time\_3): (3, 1) --> (4, 2) --> (2, 3) after loadTable currentTime is 3 ProcUsed:3 currentTime:3 Time: 0 1 2 3
12 13 14 15 16
25 26 27 28 22 17 18 20 19 21 23 24 4 Proc:3 | 5 | 11 | 11 | 11 | \_\_\_\_\_\_ Open(Time\_3): (2, 3)
Open(Time\_4): (2, 3)
In scheduling() after loadOpen currentTime is 4 Open(Time\_4): (2, 3) after loadTable currentTime is 4 ProcUsed:3 currentTime:4 Time: 0 1 2 3 4
12 13 14 15 16 17
25 26 27 28 18 19 20 21 22 24 Proc:1 | 13 | 12 | 12 | 3 | 2 Proc:2 | 10 | 1 | 1 | 4 | 4 | ------Open(Time\_4): Open (Time 5): In scheduling() after loadOpen currentTime is 5 Open (Time 5): after loadTable currentTime is 5 -----ProcUsed:3 currentTime:5 Time: 0 1 2 3 4 5 6
12 13 14 15 16 17 18 19
25 26 27 28 11 Proc:1 | 13 | 12 | 12 | 3 | 2 | 2 | Proc:2 | 10 | 1 | 1 | 4 | 4 | 0 | Proc:3 | 5 | 11 | 11 | 11 | 11 | 0 Open (Time 5): Open (Time\_6): In scheduling() after loadOpen currentTime is 6 Open (Time 6): after loadTable currentTime is 6

rime: 12 25	0 13 26	1 14 27	2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
			12				2	   			 	
Proc:2	10	1	1	4	4	0	0	 				
			11				I 0	   				
Dpen(Time_ after load	7): ing() afte 7): (7, 1) Table curr	> (6, entTime i	is 7									
	currentI		2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
			12			2		l 7				
roc:2	10	1	1	4	4		0	l 6				
Proc:3	5	11	11	11	11	1 0	0	I 0	1			
pen(Time_ fter load	8): ing() afte 8): (9, 3) Table curr current1	entTime i				5 18		7 20	8 21	9 22	10 23	11 24
roc:2								6 			 	
pen(Time_ fter load	9): ing() afte 9): Table curr	entTime i						7	8	9		11
2 5	13 26	14 27	15 28	16	17	18	19	, 20 	21	22	23	24
roc:1	13	12	12	3	2	2	2		9	9		
roc:2	10	1	1	4	4	0	1 0	l 6	1 6	6	1	
roc:3	1 5	11		11				I 0			   	
pen(Time_ fter load	10): ing() afte 10): (14, Table curr	2) entTime i										
	currentT 0 13 26		2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
roc:1	13	12	12	3	2	2	2	   7	   9	   9	   9	1

																						14		
roc:3	l 	5		11	l 	11	. I	11		11	1	0	l 	0		0	I	0	 	0	 	0	I	
pen (Tim pen (Tim n sched pen (Tim fter lo	ne_11) luling ne_11) badTak	: g() aft : (8, ole cur	2) rentT	ime i	is 11																			
====== rocUsed									====			====					====				===			
ime: 2		0 13	1			2 15		3 16		4 17		5 18		6 19		7 20		8 21		9 22		10 23		11 24
:5		26		.7		28																		
roc:1	l	13	l	12	I	12	I	3	 I	2	I	2	I	2		7		9		9	 I	9	I	8
roc:2		10		1	1	1	ı	4	I	4	I	0	ı	0		6		6	 I	6	 I	14	ı	1
roc:3	1	5		11	ı	11	ı	11		11	I	0		0		0	ı	0	 I	0	 I	0	1	0
non (Tim	4.0																							
fter lo ====== rocUsed ime: 2		ole cur ====== current 0 13	Time:	12		2		3	====	4 17		5 18	====:	6 19		7 20		8 21	====	9 22	====	10 23		11
fter lo ====== rocUsed ime: 2	adTal	ole cur ====== current 0	Time:	12		2														9		10		11
fter lo	oadTak	ole cur ====== current 0 13	Time:	12		2 15 28		16		17								21		9 22		10		11 24
fter 1c	oadTak ===== 1:3 (	ole cur  current 0 13 26	Time:	12		2 15 28		16  3 		17	   	18		19		20		21  9	  	9 22  9		10 23		11 24
fter lo	oadTah	0 13 26 13	Time:	12 4 27 12		2 15 28  12	 	16  3 	   	17 2	   	18		2 0		20  7		21  9	 	9 22 9 6	     	10 23 9	 l	11 24 8
fter lo		ble curent 0 13 26 10 10 5	Table	12 4 27 		2 15 28  12 1	 	16 3 3 4	   	17 2 4	   	18	 	19	l	7 6	I	9 6	     	9 22 9 6	   	9	     	11 24 8 
ppen(Timen main() main(		ble curent of the contract of	Table	12 4 7 7 12 12 11 11 11 13 4 4 4	             	2 15 12 11 11	 	16 3 3 4	 	17 2 4	   	18	 	19	       	7 6	I	9 6	     	9 22 9 6	   	9	     	111 24 8  0
fter lo		to ble current of the	Table	12 4 77 12 11 11 11 13 4 77	l	12 11 11 11 22 115 228		111 33 116	 	11		2 0 0 0 5 18 18 18 18 18 18 18 18 18 18 18 18 18	 	0 0 0 6 19	 	20 7 6 0 7		9 0 0		9 22 9  6 0		10 23 9 14 0	1	111 24 8 8 10
roc:1 8 roc:1 8 roc:3 0 roc:3 0 roc:3		to ble current of the	Table	12 4 77 12 11 11 11 13 4 77	l	12 11 11 11 22 115 228		111 33 116	 	11		2 0 0 0 5 18 18 18 18 18 18 18 18 18 18 18 18 18	 	0 0 0 6 19	 	7 0 0 7 20		9 0 0		9 22 9  6 0		10 23 9 14 0	1	111 24 8 8 1. 0
fter lo		ble current 0 13 26 13 10 13 26 13 10 13 26 13 10 10 11 11	Table	12 4 7 7 12 11 11 11 13 4 4 7 7 12 12 12 12 12 12 12 12 12 12 12 12 12	l	2 2 115 12 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		11 3 3 1163 3 3		11		18 2 0 0		0 0 0 6 19 2 2		7 0 0 7 20 7 7 20 7 7		9 6 0 0 8 8 21 9 9		9 22 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		10 23 9 14 0	1	111 24 8 8

# \*\*deBugFile\*\*

Entering setMatrix method!
Entering countParent() method!
In countParent() parent count is 0

Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 3 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 5 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 4 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 4 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 0 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2

```
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 2
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 0
Leaving countDependent() method.
Before leaving setMatrix method, the Matrix below
14 0 1 1 1 0 5 4 4 1 0 0 0 0 2
3 1 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 0 0 0
2 0 0 0 0 1 0 1 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0
2 0 0 0 0 0 1 0 0 0 1 0 0 0 1
2 0 0 0 0 0 0 1 1 0 0 1 0 0 0
1 0 0 0 0 0 0 1 0 0 1 0 0
1 0 0 0 0 1 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1
Leaving serMatrix().
Entering loadJobTimeAry() method!
5
6
8
9
12
13
15
18
19
23
25
26
Leaving loadJobTimeAry() method.
Entering scheduling() method!
Entering loadOpen() method!
Find an orphan, the orphan is 1
Find an orphan, the orphan is 5
Find an orphan, the orphan is 10
Find an orphan, the orphan is 11
Find an orphan, the orphan is 12
Find an orphan, the orphan is 13
Before leaving loadOpen().
Open(Time_0): (13, 1) \longrightarrow (10, 1) \longrightarrow (5, 1) \longrightarrow (12, 2) \longrightarrow (1, 2) \longrightarrow (11, 4)
Leaving loadOpen() method.
```

```
Open(Time 0): (13, 1) --> (10, 1) --> (5, 1) --> (12, 2) --> (1, 2) --> (11, 4)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
______
ProcUsed:3 currentTime:0
                            5
Time: 0 1 2
                   3
                                     7
                   16 17 18 19
12
      13
          14
               15
                                     20
                                          21
          27
2.5
      26
              28
______
______
Proc:1 | 13 |
______
______
______
______
______
Open (Time 0): (12, 2) \longrightarrow (1, 2) \longrightarrow (11, 4)
Open (Time_0): (12, 2) --> (1, 2) --> (11, 4)
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
11 0 1 1 1 0 2 4 4 1 0 0 0 0 1
3 2 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0
2 0 0 0 0 0 1 0 1 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0
2 0 0 0 0 0 1 0 0 0 0 0 0 1
2 0 0 0 0 0 0 1 1 0 0 2 0 0 0
1 0 0 0 0 0 0 0 1 0 0 0 2 0 0
1 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1
Entering loadOpen() method!
Before leaving loadOpen().
Open (Time_1): (12, 2) --> (1, 2) --> (11, 4)
Leaving loadOpen() method.
```

Entering loadTable() method!

```
Entering loadTable() method!
Open (Time 1): (12, 2) \longrightarrow (1, 2) \longrightarrow (11, 4)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
Leaving getNextProc() method. No available proc can be found.
Before leaving loadTable().
______
______
ProcUsed:3 currentTime:1
Time: 0 1 2
                                      7
                   16 17 18
12
      13
          14
               15
                                 19
                                      20
                                          21
     26
          27
2.5
              28
______
Proc:1 | 13 | 12 |
______
Proc:3 | 5 | 11 |
______
______
Open (Time 1):
Open(Time_1):
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Leaving deleteDoneJobs() method below is the updated matrix.
11 0 1 1 1 0 2 4 4 1 0 0 0 0 1
3 2 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0
2 0 0 0 0 0 1 0 1 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 0 0 1 1 0 0 0 0
2 0 0 0 0 0 0 1 1 0 0 2 0 0 0
1 0 0 0 0 0 0 1 0 0 0 2 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1
Entering loadOpen() method!
Before leaving loadOpen().
Open (Time 2):
Leaving loadOpen() method.
Entering loadTable() method!
Open (Time 2):
***Set 4 with unlimited processors***
**OutFile1**
```

In main() after loadMatrix().

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 1
In main() after setMatrix().
14 0 1 1 1 0 5 4 4 1 0 0 0 0 2
2 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 0 0 0 0
2 0 0 0 0 0 1 0 1 0 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0 0
20000011001000
  0 0 0 0 0 0 1 0 0 0 1 0 0
in main() after allocate and initialize table.
______
ProcUsed:0 currentTime:0
                 2
   0 1
13 14
Time:
                                                                  11
12
                 15
                       16
                            17
                                 18
                                       19
                                            20
                                                  21
                                                       22
                                                            23
                                                                  24
                28
25
      26
            27
      -----
Proc:1 | 0 |
Proc:2 | 0 |
Proc:3 | 0 |
Proc:4
     | 0
Proc:5 | 0
Proc:6 | 0
        1
Proc:7 | 0 |
Proc:8
    | 0
     1 0
Proc:9
Proc:10 | 0 |
Proc:12 | 0 |
Proc:13 | 0 |
Proc:14 | 0 |
Open (Time 0):
In scheduling() after loadOpen currentTime is 0
Open(Time_0): (13, 1) --> (10, 1) --> (5, 1) --> (12, 2) --> (1, 2) --> (11, 4) after loadTable currentTime is 0
______
ProcUsed:6 currentTime:0
Time: 0 1 2
12 13 14 15
25 26 27 28
                     3 4
16 17
                                 5 6 7 8
18 19 20 21
                                                       22
                                                            23
                                                                  24
```

	13	ı										
Proc:2	10	1										
Proc:3		1										
	    12											
Proc:6												
Proc:7		1										
Proc:8		l										
		1										
Proc:10		l										
Proc:11		1										
Proc:12		l										
Proc:13		l 										
Proc:14	0	1										
Open (Time_												
Open(Time_ after load	1): Table curr	entTime	en currentT is 1									
Open (Time_after load ====================================	1): Table curr current1	rentTime	is 1 ====================================	 	4	<b></b> 5	6	7	8	9	10	11
Open (Time_ after load ======= ProcUsed:6	1): Table curr ======= current1	centTime	is 1 ====================================									
Open (Time_after load ====================================	1): Table current  current  0  13  26	rentTime  	is 1 ====================================	 	4	<b></b> 5	6	7	8	9	10	11
Open(Time_after load ====================================	1): Table current  current  0  13  26	rentTime ====================================	2 15 28	 	4	<b></b> 5	6	7	8	9	10	11
Open(Time_after load ====================================	1): Table curr	PentTime: 1 1 1 1 27	1s 1  2 15 28	3 16	4	<b></b> 5	6	7	8	9	10	11
Open (Time after load	1): Table curr	PrentTime	2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr	Control   Cont	2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Current7 0 13 26   13   15   10   12   12	Control   Cont	is 1  2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Current7 0 13 26   13   10   10   10   11   12   11	Control   Cont	is 1  2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Current7 0 13 26 1 13 1 10 1 12 1 12 1 1	Continue	2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Current7 0 13 26   13   10   10   10   11   10   1 1   1 1   1 0   1 1   1 0	Continue	2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Current 10 13 26	CentTime	is 1  2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Current 10 13 26 11 13 11 10 11 10 11 12 11	CentTime	is 1  2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Table current  0 13 26   13   10   10   10   0   0   10   0   0   0   0   0   0	Continue	is 1  2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24
Open (Time after load	1): Table curr  Table current  0 13 26   13   10   10   5   12   11   11   0   0   0   0   0   0	Continue	is 1  2 15 28	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	11 24

```
______
Open(Time_1):
Open (Time_2):
Open(Time_2): (3, 1) --> (4, 2) --> (2, 3) after loadTable currentTime is 2
ProcUsed:6 currentTime:2
Time:
                       4
17
     13
                            18
          14
27
                                19
                                     2.0
                                              2.2
12
25
              1.5
                  16
                                         21
                                                       2.4
              28
   | 13
        1 0
             | 3
Proc:2
    1 10
         1 0
             1 4
______
Proc:3
______
         | 12
Proc:4
    | 1
         | 1
              0
        | 11
             | 11
    | 11
Proc:6
         | 0
             | 0
Proc:7
    1 0
        | 0
             | 0
Proc:9
    1 0
       I 0
            | 0
        1 0
              0
    1 0
             Proc:10
                  0
Proc:11
          Ω
               0
Proc:13
    1 0
         1 0
              0
Proc:14 | 0 | 0 | 0
______
Open(Time_2):
Open(Time_3):
In scheduling() after loadOpen currentTime is 3
Open(Time_3):
after loadTable currentTime is 3
______
ProcUsed:6 currentTime:3
Time:
      0
                                                       11
     13
12
              15
                   16
                       17
                            18
                                 19
                                     20
                                         21
                                              22
                                                       24
25
     26
          27
              28
______
             1 3
   1 13
        1 0
                 1 0
Proc:1
                      ______
Proc:2
      10
               4
                   4
          0
               2
                   2
Proc:4
    1 12
          12
               Ω
                  1 0
    1 1
         1
             1 0
                  1 0
Proc:5
             | 11
                  | 11
    | 11
Proc:6
         | 11
Proc:7
    1 0
         1 0
             1 0
                  1 0
            | 0
   | 0
        | 0
Proc:8
                 1 0
                      - 1
```

Proc:9		0	I 0	   	0		0	I						 				 	
Proc:10	   	0	1 0	   	0		0	1						 				 	
Proc:11	   	0	I 0	   	0	 	0	I						 				 	
Proc:12	 	0	I 0	 	0	I	0	 I						 				 	
Proc:13	- 1	0	I 0	   	0	 	0	 						 				 	
Proc:14	 	0	I 0	   	0	 	0	 						 				 	
Open (Time Open (Time In sched Open (Time after loc	e_4) ulin e_4) adTa	: g() afte : ble curi	rentTime	is 4										 				 	
ProcUsed														 					
Time: 12 25		0 13 26	1 14 27		2 15 28		3 16		4 17		5 18		6 19	7 20	2	1	9 22	10 23	11 24
Proc:1	 	13	I 0	 	 3	 	0	 I	0	 I				 				 	
Proc:2	 	10	0 	 	4	 	4	I	0					 				 	
Proc:3	 		I 0	 	2 	 	2	I	2					 				 	
Proc:4	I		12	I	0	I	0	1	0	1				 				 	
Proc:5		1	1		0		0	ı	0	ı				 				 	
Proc:6		11	11	   	11		11	1	0	1				 				 	
Proc:7	   	0	I 0	 	0		0	ı	0	ı				 				 	
Proc:8		0	I 0	   	0	 	0	1	0					 				 	
Proc:9	   	0	I 0	   	0		0	ı	0					 				 	
Proc:10	   	0	I 0	   	0	 	0	I	0	 				 				 	
Proc:11	   	0	I 0	   	0	 	0	I	0	 				 				 	
Proc:12	 	0	I 0	   	0	 	0	1	0	 				 				 	
Proc:13			I 0	 		 			0	l				 				 	
Proc:14	- 1	0	I 0	1	0	1	0	- 1	0	- 1				 				 	
Open (Time Open (Time In sched Open (Time after loa	e_4) e_5) ulin e_5) adTa	: : g() afte : (7, 1) ble curi	er loadOj )> (6 rentTime	pen c	urren	tTime	e is 5							 					
ProcUsed Time: 12 25		====== current1							4 17		5 18		6 19	 7 20	8 2	1	9 22	 10 23	11 24
Proc:1			0				0			- 1	7	1						 	
Proc:2	   	10	0	   	4	I	4		0		6	 		 				 	
Proc:3		5	1 0	 	2		2	1	2		0			 					

Proc:4	 	12	 	12	 	0 	 	0	 	0 		0	 			 			
	1		1	1		0		0		0		0	I			 			
	1	11	ı			11				0		0				 			
Proc:7			1	0	   	0	 	0	 	0	 	0	 			 			
Proc:8	1	0	1	0		0		0	I	0	 	0	I			 			
Proc:9	 	0		0	 I	0	 I	0		0	 	0	 I			 			
Proc:10	 	0	 	0	 	0	 	0	 I	0	 	0	 			 			
Proc:11	 	0	 	0	 	0	 	0	 	0	 	0	 			 			
  Proc:12	 	0	 	0		 0	 	0	<u>-</u> - 	0	 	0				 			
	 	0 	<u> </u>	0 	 	0 			 	0 	l	0	 			 			
Proc:14			1	0	 	0		0		0	 	0	I			 			
	6): ling 6): dTak	: g() afte : (9, 3) ole cur:	) rent ==== Time	tTime :	is 6												9 22		11 24
 Proc:1	 	13		 0		 		 0	 I	 0		7	 	9	 	 			
					 					 			<u>-</u> -			 			
Proc:2	 	10	 	0 		4		4	 	0 		6 	I	6 	I	 			
Proc:3				0		2		2		2		0				 			
Proc:4	1	12	1	12		0		0		0	I	0		0		 			
Proc:5	I	1	ı	1		0		0		0		0	ı	0		 		· 	
Proc:6	 	11	ı	11	l	11		11	 	0	   	0	 	0		 			
Proc:7	1	0	1	0	 	0		0	 	0	   	0		0		 			
Proc:8						0		0				0		0					
Proc:9	1	0	1	0	1	0	- 1	0	- 1	0	- 1	0	1	0	1				
Proc:10																 			
Proc:11								0					1	0	1				
Proc:12	 	0		0	 I	0			 I	0		0	 I	0	I	 			
Proc:13	 	0		0	 I	0	- 1	0	 I	0		0	 I	0	I				
	6): 7): Ling	: : g() afte								<del>-</del>						 			

### PROOF	11 24	10 23	9 22	8 21	7 20		6 19		5 18		4 17		3 16	:	2 15		9:7 1 14	tTime	current 0 13	Used:6
Tree:13   10   1 0   1 4   1 4   1 0   6   6   6   6   7   7   7   7   7   7															28		27 		26 	
TOOLS		 		l	9		9	   	7	l	0		0	l	3		0		13	:1
roci3   5   0   1   2   1   2   1   2   1   0				1		 	6	 	6		0	 I	4	l	4	 	0	 	10	:2
roc:5   1				 	0	 	0	 	0	 	2	I	2	l	2	 	0	 I	5	::3
rec:6   11   11   11   11   11   11   10   10   0				 I	0	 I	0	 I	0	 I	0		0	 	0	 	12	 I	12	:4
Table 1				 	0	 	0	 	0	 	0	I	0	l	0	 	1	 I	1	::5
cocis   0				 	0	I	0	 I	0		0	 I	11		11	ı	11		11	:6 I
COC120   0   0   0   0   0   0   0   0   0				 I	0	 I	0	 	0	 	0		0	 	0	 I	0	 I	0	::7
roc:10   0   0   0   0   0   0   0   0   0				  I	0	 I	0	 	0	 	0		0	 	0	 I	0	 	0	:8
roc:11   0   0   0   0   0   0   0   0   0				  I	0	 	0	 	0	 	0	 	0	 	0	 	0	 	0	::9
Troc:12   0   0   0   0   0   0   0   0   0				  I	0	 	0	 	0	 	0	 	0	 	0	 	0	 	0	::10
roc:13   0   0   0   0   0   0   0   0   0				  I	  0	 	 	 	 0	 	0		 0	 	 0	 	 	 	  0	:11
roc:13   0   0   0   0   0   0   0   0   0						<u>-</u>	 	<u>-</u> -	 	<u>-</u> -			 		 	<u>-</u>	 	<u>-</u>		
roc:14   0   0   0   0   0   0   0   0   0		 		1	 	 	 		 							 		 	·	
pen(Time_7): pen(Time_8): n scheduling() after loadOpen currentTime is 8 pen(Time_8): (14, 2) tter loadTable currentTime is 8		 		   		 	 		 	<u>-</u> -			 	<u>-</u>	 		 		 	::13
roc:1	11 24	10	9	8	7	====	6	====	5		4		3		2 15	is 8	tTime : ====== =:8 1 14	2) rrent ===== tTime	g() aft : (14, bble cur 	chedulir (Time_8) r loadTa
roc:2    10    0    4    4    0    6    6    6							 		 				 		 					
roc:3   5   0   2   2   2   0   0   0   0   0   0		 				 								<u>-</u>		 	 	<u>-</u>		 
roc:4   12   12   0   0   0   0   0   0   0   0   0																				
roc:5   1   1   0   0   0   0   0   0   0   0																				
roc:6   11   11   11   11   0   0   0   0   0																				
roc:7   0   0   0   0   0   0   0   0   0																				
roc:7   0   0   0   0   0   0   0   0   0																				
roc:8   0   0   0   0   0   0   0   0   0																				
			l 	0	0	<u> </u>	0		0	I	0		0	<u> </u>	0		0		0	:8
			l 	l 0 	0	 	0	 	0	I	0	I	0	I	0		0	I	0	:9
roc:10   0   0   0   0   0   0   0   0   0		·- ·	1	0	0	1	0	- 1	0	1	0	- 1	0	1	0	1	0	- 1	0	::10

Proc:12		0	I 0		0	I 0	l 0	l 0	0	0	0	 		
roc:13		0	1 0		0	J 0	J 0	J 0	0	J 0	0	l	-	
roc:14		0	1 0	 	0	I 0	I 0	I 0	l 0	I 0	I 0	 	-	
pen(Time fter lo	e_9) ulin e_9) adTa ====	: g() aft : (8, 2	rentTime			Time is 9								
rocUsed ime:					2	3	4	5	6	7	8	9	10	11
.2 ?5		13 26	14 27		15 28	16	17	18	19	20	21	22	23	24
roc:1		13	0	I	3	I 0	I 0	l 7	l 9	J 9	l 9	l 8	- - I	
roc:2	 	10	0	 	4	4	0	6	6	6	14	14	-   	
roc:3	 	5	0	I	2	2	2	I 0	I 0	I 0	I 0	I 0	-   	
Proc:4	 I	12	12	I	0	I 0	I 0	I 0	l 0	I 0	I 0	I 0	-	
Proc:5	 	1	1	 I	0	I 0	0	I 0	I 0	I 0	I 0	I 0		
Proc:6	 		11	 	11	11	I 0	I 0	I 0	I 0	I 0	I 0	 - 	
?roc:7			I 0	 I		I 0	I 0	I 0	I 0	I 0	I 0	I 0	- - 	
roc:8		0	I 0	 I	0	I 0	I 0	I 0	I 0	0	I 0	I 0	- - 	
roc:9	 	0	0	I	0	I 0	0	I 0	I 0	0	I 0	I 0	- - 	
Proc:10	 	0	I 0	 	0	l 0	l 0	I 0	I 0	I 0	I 0	I 0	-   	
Proc:11	 	0	0	 	0	I 0	0	I 0	I 0	I 0	I 0	I 0	-   	
Proc:12	   	0	I 0	 	0	I 0	I 0	I 0	l 0	0	I 0	I 0	-   	
roc:13	 	0	1 0	 	0	I 0	I 0	I 0	0	0 	I 0	I 0	-   	
roc:14	 	0	I 0	 	0	I 0	I 0	I 0	0	I 0	I 0	0	-   	
Open(Time after loa	e_10 ulin e_10 adTa ====	): g() aft ): ble cur	rentTime	is 1	10									
rime: 12		0	1 14		2 15	3 16	4 17	5 18	6 19	7 20	8 21	9 22	10 23	
									9 					
roc:2		10	I 0	1	4	4	I 0	6		6	14	14	I 0	l 
roc:3		5	1 0		2	2	2	I 0	I 0	I 0	I 0	I 0	I 0	I
roc:4		12	12	1	0	0	0	1 0	I 0	0	0	0	0	
			1						l 0				-   0	l 
													_	

Proc:7		0	 	0		0	1	0		0		0		0	 	0		0	 	0	 	0	I	
Proc:8	ı	0	 	0	ı	0	 	0	ı	0	l	0		0	   	0		0		0	1	0	ı	
Proc:9	ı	0	 	0	l	0		0	l	0	ı	0	ı	0	 	0		0	 	0	 	0	ı	
 Proc:10	1	0	 I	0	 I	0		0	 I	0		0		0		0		0	 I	0		0	1	
 Proc:11		0	 	0	 	0	 	0	 		 		 	0	 	0	 	0	 	0	  	0	·	
  Proc:12		0			 		 			0	 				 		 		 			0	 I	
					 		<u>-</u> -						 		<u>-</u>		<u>.</u>							
Proc:13 	 				 		<u>-</u>														 			
Proc:14 	 	0	 	0 	 	0	 	0 	 	0	 	0 	 	0	 	0	 	0 	 	0 	 	0		
Open(Time Open(Time In main()	=_11) ) pri =====	: Inting																						
ProcUsed: Pime:	:6 0		Time		====	2		3		4		5		6		7		8	====	9	==	10		11
12 25 		13 26		14 27		15 28		16		17		18		19		20		21		22		23		24
 Proc:1 	ı	13	   	0	   	3		0		0		7				9			 I	8	 I	8	l 	0
?roc:2	I	10	   	0	   	4	l	4	ı	0	l	6	I	6	l	6	l	14	   	14	   	0	1	0
?roc:3	ı	5	   	0	 	2	 	2	ı	2		0	 	0	   	0	   	0	   	0	   	0	1	0
I	I		I	12	1	0	I	0	1	0	I	0	I	0	I	0	I	0	I	0	I			0
 Proc:5 	ı			1	 	0	I	0	ı	0		0	 I	0	 I	0	l	0	 I	0	 		ı	0
 Proc:6	ı	11	   	11	l	11	l	11	1	0	l	0	I	0	   	0	l	0	 I	0	 I	0	ı	0
Proc:7	I	0	ı	0	l	0	l	0	l	0	l	0	ı	0	   	0	l	0	 	0	 I	0	ı	0
Proc:8	I									0										0	'	Ü	'	
?roc:9	ı			0	 I	0	 I	0	ı	0	 I	0	 I	0	 l	0	Ι	0	 	0	<sub> </sub>	0	1	0
																						0		0
Proc:10	I																							
Proc:10	 	0	 	0	 	0	 	0		0	l	0	ı	0	 l	0	l	0	 	0	 	0	ı	0
Proc:10	 	0	     	0	   	0	     	0	 	0	 	0	 	0	 	0	 	0	 	0	    	0	 	0  0
Proc:10 Proc:11 Proc:12 Proc:13	 	0	     	0	 	0	 	0		0	 	0	 	0 0	      	0		0	       	0	        	0	 	0 0 0

#### \*\*deBugFile\*\*

Entering setMatrix method! Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 3 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 5 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 4 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 4 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 0 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 1 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 1 Leaving countDependent() method. Entering countParent() method! In countParent() parent count is 0 Leaving countParent() method. Entering countDependent() method! In countDependent() dependent count is 2

```
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 2
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 0
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 1
Leaving countDependent() method.
Entering countParent() method!
In countParent() parent count is 2
Leaving countParent() method.
Entering countDependent() method!
In countDependent() dependent count is 0
Leaving countDependent() method.
Before leaving setMatrix method, the Matrix below
14 0 1 1 1 0 5 4 4 1 0 0 0 0 2
3 1 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 1 0 0 0 0 0 0 0
2 0 0 0 0 0 1 0 1 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 0 0
1 0 0 0 0 0 0 0 1 1 0 0 0 0
2 0 0 0 0 0 1 0 0 0 1 0 0 0 1
2 0 0 0 0 0 0 1 1 0 0 1 0 0 0
1 0 0 0 0 0 0 1 0 0 0 1 0 0
1 0 0 0 0 1 0 0 0 0 0 1 0
Leaving serMatrix().
Entering loadJobTimeAry() method!
5
6
8
9
12
13
15
18
19
23
25
26
Leaving loadJobTimeAry() method.
Entering scheduling() method!
Entering loadOpen() method!
Find an orphan, the orphan is 1
Find an orphan, the orphan is 5
Find an orphan, the orphan is 10
```

```
Find an orphan, the orphan is 11
Find an orphan, the orphan is 12
Find an orphan, the orphan is 13
Before leaving loadOpen().
Open(Time_0): (13, 1) --> (10, 1) --> (5, 1) --> (12, 2) --> (1, 2) --> (11, 4)
Leaving loadOpen() method.
Entering loadTable() method!
Open(Time_0): (13, 1) --> (10, 1) --> (5, 1) --> (12, 2) --> (1, 2) --> (11, 4)
Entering getNextProc() method!
in getNextProc() found an available proc#1
Entering getNextProc() method!
in getNextProc() found an available proc#2
Entering getNextProc() method!
in getNextProc() found an available proc#3
Entering getNextProc() method!
in getNextProc() found an available proc#4
Entering getNextProc() method!
in getNextProc() found an available proc#5
Entering getNextProc() method!
in getNextProc() found an available proc#6
Entering getNextProc() method!
in getNextProc() found an available proc#7
Before leaving loadTable().
______
_____
_____
ProcUsed:6 currentTime:0
Time: 0 1 2
               3
                  4
                      5
                         6
                              7
                                        1.0
                                            11
12
    13
        14
           15
               16
                  17
                      18
                         19
                             20
                                 21
                                        23
                                            24
                                    2.2
25
______
______
______
Proc:2 | 10 |
______
______
Proc:3 | 5 |
_____
Proc:4 | 12 |
______
______
Proc:5 | 1 |
_____
Proc:6 | 11 |
______
Proc: 7 | 0 |
_____
Proc:8 | 0 |
```

```
______
_____
______
    | 0 |
Proc:10
______
______
______
     | 0 |
Proc:11
______
     | 0 |
______
______
______
     | 0 |
Proc:13
     | 0 |
______
______
Open(Time 0):
Open (Time 0):
Leaving loadTable() method.
Entering deleteDoneJobs() method!
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Entering deleteOneJob() method!
Leaving deleteOneJob() method.
Leaving deleteDoneJobs() method below is the updated matrix.
11 0 1 1 1 0 2 4 4 1 0 0 0 0 1
3 2 1 1 1 0 0 0 0 0 0 0 0 0 0
2 0 1 0 0 0 1 1 0 0 0 0 0 0
2 0 0 1 0 0 1 1 0 0 0 0 0 0
1 0 0 0 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0
2 0 0 0 0 0 1 0 1 0 0 0 0 0 1
1 0 0 0 0 0 0 1 0 1 0 0 0 0
1 0 0 0 0 0 0 1 1 0 0 0 0
2 0 0 0 0 0 1 0 0 0 0 0 0 1
20000011002000
1 0 0 0 0 0 0 1 0 0 0 2 0 0
1 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 1
Entering loadOpen() method!
Before leaving loadOpen().
Open(Time 1):
Leaving loadOpen() method.
Entering loadTable() method!
Open (Time 1):
```