



**North South University**

**Department of Electrical and Computer Engineering**

## **PROJECT REPORT**

**CSE323**

**Operating System Design**

**Section- 5**

### **Deadlock Solution**

**Group Members:**

<b>Akmam Hasan</b>	<b>-- 1911192042</b>
<b>Rafi Majid</b>	<b>-- 1922111642</b>
<b>Kazi Rabiul Alam</b>	<b>-- 1921335042</b>

**Date of Submission: 12.01.2022**

## **Deadlock**

Deadlock occurs in operating system when a process or thread enters a waiting state. Deadlock occurs because requested system resource is held by another waiting process, which in turn is waiting for another resource, held by another waiting process.

## **Banker's Algorithm**

Banker's Algorithm is resource allocation and deadlock avoidance algorithm which test all the request made by processes for resources, it checks for the safe state, if after granting request system remains in the safe state it allows the request and if there is no safe state it doesn't allow the request made by the process.

## **Resource Allocation Graph**

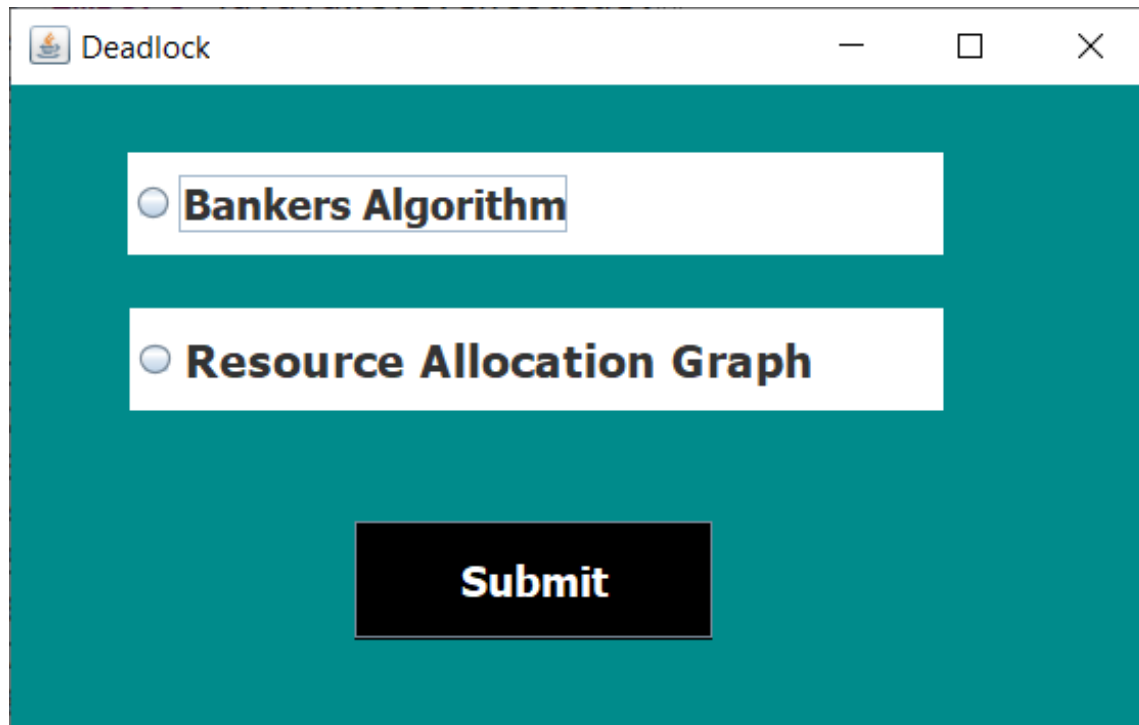
The resource allocation graph is the pictorial representation of the state of a system.

## **About Project**

In this project, we used Banker's Algorithm and Resource Allocation Graph for deadlock avoidance. Banker's Algorithm is resource allocation and deadlock avoidance algorithm. We have used Java programming language for the development. For GUI implementation we have used Swing which also based on Java. This project contains 3 java classes, for home window, Banker's algorithm and RAG.

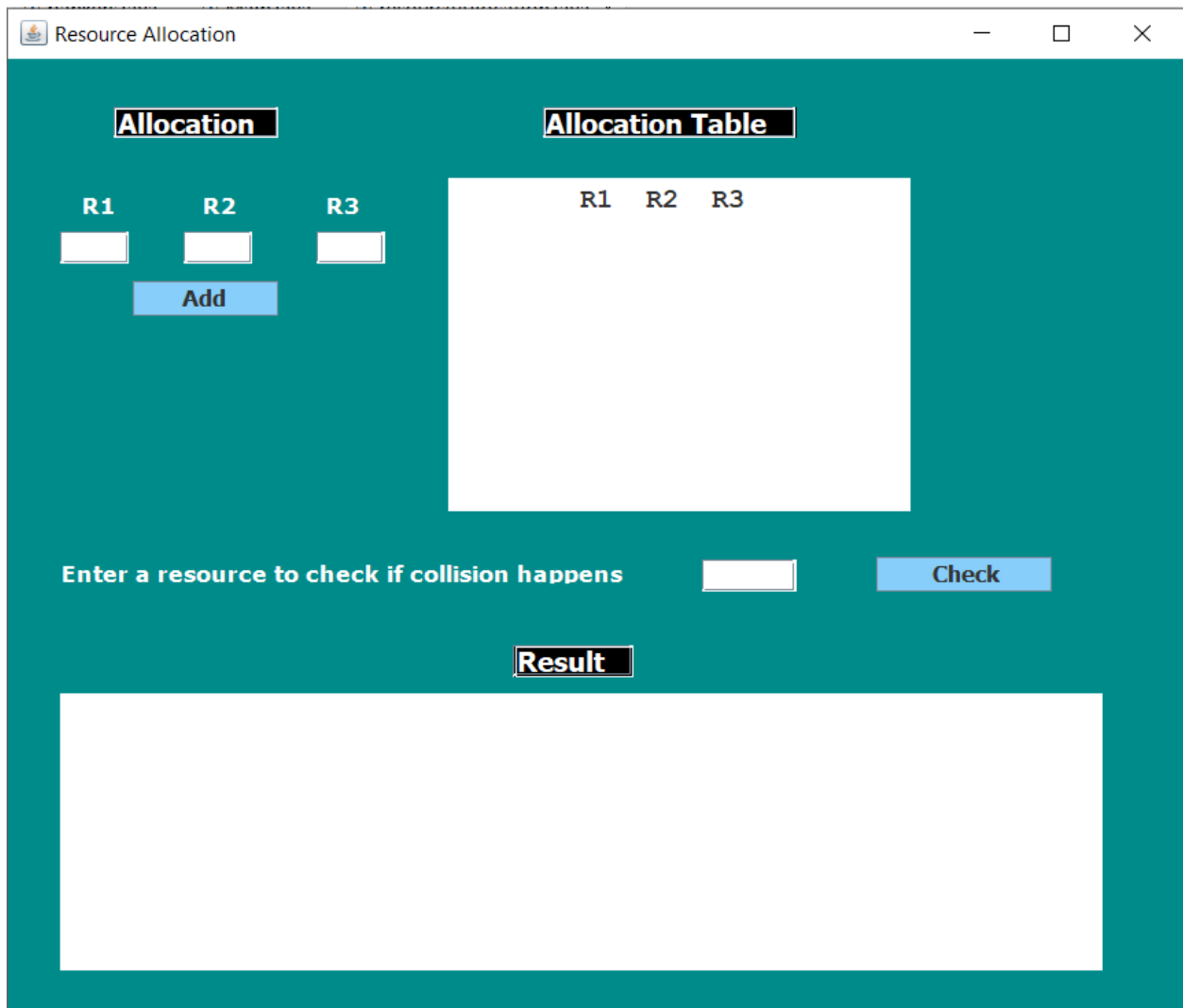
We also have created two separate interfaces, one for Banker's algorithm and another for RAG. For interface, we have used JAVA Swing.

When we execute the program, we would be able to see a interface. In the interface there are two option to select the algorithm. By selecting each we would be able to see the functionality of each program.



The image shows a screenshot of a Java Swing window titled "Deadlock". The window has a teal background. It contains two radio buttons, each with a white circular indicator. The first radio button is selected and is followed by the text "Bankers Algorithm". The second radio button is also selected and is followed by the text "Resource Allocation Graph". Below these two options is a black rectangular button with the word "Submit" in white text.

If we select Resource Allocation Graph, then this window will open.



The image shows a window titled "Resource Allocation" with a teal background. It contains several sections:

- Allocation:** Three input fields labeled R1, R2, and R3, each with a small white box. Below them is a blue button labeled "Add".
- Allocation Table:** A large white rectangular area with the headers R1, R2, and R3 at the top.
- Check Section:** A label "Enter a resource to check if collision happens" followed by a white input field and a blue button labeled "Check".
- Result:** A large white rectangular area at the bottom.

We can allocate resources and also enter a resource to check for collusion. The test area allocation table will show the allocated resources. After clicking on check button for collusion, the text area of Result will show the count of deadlocks.

One example is shown below:

Resource Allocation

Allocation

R1

R2

R3

Add

Allocation Table

R1	R2	R3
0	1	0
2	0	0
3	0	2
2	1	1
0	0	2

Enter a resource to check if collision happens

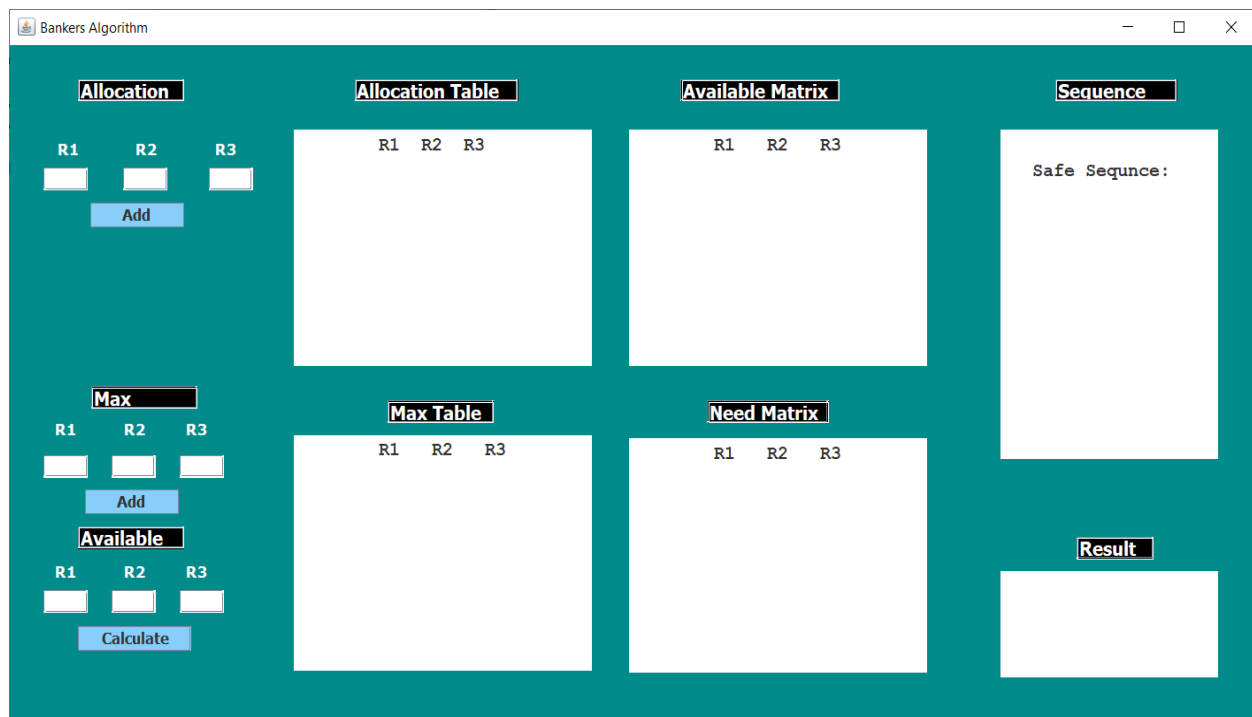
3

Check

Result

Counts of deadlock: 0

After selecting Banker's Algorithm, this window will open.



The image shows a software window titled "Bankers Algorithm" with a teal background. It contains several input fields and buttons for calculating a safe sequence. On the left, there are three sections: "Allocation" with input fields for R1, R2, and R3 and an "Add" button; "Max" with input fields for R1, R2, and R3 and an "Add" button; and "Available" with input fields for R1, R2, and R3 and a "Calculate" button. In the center, there are three large empty boxes labeled "Allocation Table", "Max Table", and "Need Matrix", each with headers R1, R2, and R3. On the right, there is a "Sequence" section with a text area labeled "Safe Sequence:" and a "Result" section with an empty text area. The window has standard minimize, maximize, and close buttons in the top right corner.

We have to add values of resources of each process separately which will be displayed on text area titled "Allocation table". Then we need to provide the max values of resources which will be display on "Max table". After giving the values of available resource we need to click on "Calculate" to see the safe sequence and result. After clicking on "Calculate" we will be seeing the value of available resources on the text area titled" Available Matrix" and the need values on text area titled "Need Matrix". The calculated safe sequence would be displayed on text area titled "Sequence" and the result would be showed on text are titled "Result".

It will show the safe sequence and detect whether there is any safe sequence or not.

An example is shown below:

Bankers Algorithm

Allocation

R1

R2

R3

Add

Max

R1

R2

R3

Add

Available

R1

R2

R3

Calculate

Allocation Table

R1	R2	R3
0	1	0
2	0	0
3	0	2
2	1	1
0	0	2

Available Matrix

R1	R2	R3
3	3	2

Max Table

R1	R2	R3
7	5	3
3	2	2
9	0	2
2	2	2
4	3	3

Need Matrix

R1	R2	R3
7	4	3
1	2	2
6	0	0
0	1	1
4	3	1

Sequence

Safe Sequence:  
Process1  
Process3  
Process4  
Process0  
Process2

Result

Safe