

# CS342 – Operating Systems Lab

## Lab 8

### Bankers Algorithm

Tarusi Mittal

1901CS65

---

**Question 1:** Write a Program to Implement Bankers algorithm.

- a) Given allocated, max and available resources (should be taken as input from user) for n number of process The code should print if there is any safe sequence available.
- b) For the above problem, print all the different safe sequences possible.

**Answer:**

**What is Bankers Algorithm?**

The banker's algorithm is a resource allocation and deadlock avoidance algorithm that tests for safety by simulating the allocation for predetermined maximum possible amounts of all resources, then makes an "s-state" check to test for possible activities, before deciding whether allocation should be allowed to continue.

**Compilation:**

```
g++ -P1.cpp -o P1
```

**Syntax:**

```
./P1  
Number of processes  
Number of resources  
Available resources array  
Allocated resources
```

## Sample Output:

```
tarusimittal@LAPTOP-6CRHF1G0:/mnt/c/Users/Tarusi Mittal/desktop/1901CS65$ g++ P1.cpp -o P1
tarusimittal@LAPTOP-6CRHF1G0:/mnt/c/Users/Tarusi Mittal/desktop/1901CS65$ ./P1
Enter the no of process: 5
Enter the no of resources: 3
Enter the available resources with spaces:
3 3 2
Enter the allocation matrix in process*resources order:
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the maximum matrix in process*resources order:
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
Safe sequence 1: 1 -> 3 -> 0 -> 2 -> 4
Safe sequence 2: 1 -> 3 -> 0 -> 4 -> 2
Safe sequence 3: 1 -> 3 -> 2 -> 0 -> 4
Safe sequence 4: 1 -> 3 -> 2 -> 4 -> 0
Safe sequence 5: 1 -> 3 -> 4 -> 0 -> 2
Safe sequence 6: 1 -> 3 -> 4 -> 2 -> 0
Safe sequence 7: 1 -> 4 -> 3 -> 0 -> 2
Safe sequence 8: 1 -> 4 -> 3 -> 2 -> 0
Safe sequence 9: 3 -> 1 -> 0 -> 2 -> 4
Safe sequence 10: 3 -> 1 -> 0 -> 4 -> 2
Safe sequence 11: 3 -> 1 -> 2 -> 0 -> 4
Safe sequence 12: 3 -> 1 -> 2 -> 4 -> 0
Safe sequence 13: 3 -> 1 -> 4 -> 0 -> 2
Safe sequence 14: 3 -> 1 -> 4 -> 2 -> 0
Safe sequence 15: 3 -> 4 -> 1 -> 0 -> 2
Safe sequence 16: 3 -> 4 -> 1 -> 2 -> 0
tarusimittal@LAPTOP-6CRHF1G0:/mnt/c/Users/Tarusi Mittal/desktop/1901CS65$ |
```

Safe sequence 1: 1 -> 3 -> 0 -> 2 -> 4  
Safe sequence 2: 1 -> 3 -> 0 -> 4 -> 2  
Safe sequence 3: 1 -> 3 -> 2 -> 0 -> 4  
Safe sequence 4: 1 -> 3 -> 2 -> 4 -> 0  
Safe sequence 5: 1 -> 3 -> 4 -> 0 -> 2  
Safe sequence 6: 1 -> 3 -> 4 -> 2 -> 0  
Safe sequence 7: 1 -> 4 -> 3 -> 0 -> 2  
Safe sequence 8: 1 -> 4 -> 3 -> 2 -> 0  
Safe sequence 9: 3 -> 1 -> 0 -> 2 -> 4  
Safe sequence 10: 3 -> 1 -> 0 -> 4 -> 2  
Safe sequence 11: 3 -> 1 -> 2 -> 0 -> 4  
Safe sequence 12: 3 -> 1 -> 2 -> 4 -> 0  
Safe sequence 13: 3 -> 1 -> 4 -> 0 -> 2  
Safe sequence 14: 3 -> 1 -> 4 -> 2 -> 0  
Safe sequence 15: 3 -> 4 -> 1 -> 0 -> 2  
Safe sequence 16: 3 -> 4 -> 1 -> 2 -> 0

```

tarusimittal@LAPTOP-6CRHF1G0:/mnt/c/Users/Tarusi Mittal/desktop/1901CS65$ g++ P1.cpp -o P1
tarusimittal@LAPTOP-6CRHF1G0:/mnt/c/Users/Tarusi Mittal/desktop/1901CS65$ ./P1
Enter the no of process: 5
Enter the no of resources: 3
Enter the available resources with spaces:
1 2 1
Enter the allocation matrix in process*resources order:
1 2 1
1 3 2
2 3 0
0 1 0
0 0 0
Enter the maximum matrix in process*resources order:
2 3 4
2 1 3
1 2 3
1 1 1
1 3 1
Safe sequence 1: 1 -> 0 -> 2 -> 3 -> 4
Safe sequence 2: 1 -> 0 -> 2 -> 4 -> 3
Safe sequence 3: 1 -> 0 -> 3 -> 2 -> 4
Safe sequence 4: 1 -> 0 -> 3 -> 4 -> 2
Safe sequence 5: 1 -> 0 -> 4 -> 2 -> 3
Safe sequence 6: 1 -> 0 -> 4 -> 3 -> 2
Safe sequence 7: 1 -> 2 -> 0 -> 3 -> 4
Safe sequence 8: 1 -> 2 -> 0 -> 4 -> 3
Safe sequence 9: 1 -> 2 -> 3 -> 0 -> 4
Safe sequence 10: 1 -> 2 -> 3 -> 4 -> 0
Safe sequence 11: 1 -> 2 -> 4 -> 0 -> 3
Safe sequence 12: 1 -> 2 -> 4 -> 3 -> 0
Safe sequence 13: 1 -> 3 -> 0 -> 2 -> 4
Safe sequence 14: 1 -> 3 -> 0 -> 4 -> 2
Safe sequence 15: 1 -> 3 -> 2 -> 0 -> 4
Safe sequence 16: 1 -> 3 -> 2 -> 4 -> 0
Safe sequence 17: 1 -> 3 -> 4 -> 0 -> 2
Safe sequence 18: 1 -> 3 -> 4 -> 2 -> 0
Safe sequence 19: 1 -> 4 -> 0 -> 2 -> 3
Safe sequence 20: 1 -> 4 -> 0 -> 3 -> 2
Safe sequence 21: 1 -> 4 -> 2 -> 0 -> 3
Safe sequence 22: 1 -> 4 -> 2 -> 3 -> 0
Safe sequence 23: 1 -> 4 -> 3 -> 0 -> 2
Safe sequence 24: 1 -> 4 -> 3 -> 2 -> 0
Safe sequence 25: 3 -> 1 -> 0 -> 2 -> 4
Safe sequence 26: 3 -> 1 -> 0 -> 4 -> 2
Safe sequence 27: 3 -> 1 -> 2 -> 0 -> 4
Safe sequence 28: 3 -> 1 -> 2 -> 4 -> 0
Safe sequence 29: 3 -> 1 -> 4 -> 0 -> 2
Safe sequence 30: 3 -> 1 -> 4 -> 2 -> 0
Safe sequence 31: 3 -> 4 -> 1 -> 0 -> 2
Safe sequence 32: 3 -> 4 -> 1 -> 2 -> 0

```

Safe sequence 1: 1 -> 0 -> 2 -> 3 -> 4  
 Safe sequence 2: 1 -> 0 -> 2 -> 4 -> 3  
 Safe sequence 3: 1 -> 0 -> 3 -> 2 -> 4  
 Safe sequence 4: 1 -> 0 -> 3 -> 4 -> 2  
 Safe sequence 5: 1 -> 0 -> 4 -> 2 -> 3  
 Safe sequence 6: 1 -> 0 -> 4 -> 3 -> 2  
 Safe sequence 7: 1 -> 2 -> 0 -> 3 -> 4  
 Safe sequence 8: 1 -> 2 -> 0 -> 4 -> 3  
 Safe sequence 9: 1 -> 2 -> 3 -> 0 -> 4  
 Safe sequence 10: 1 -> 2 -> 3 -> 4 -> 0  
 Safe sequence 11: 1 -> 2 -> 4 -> 0 -> 3  
 Safe sequence 12: 1 -> 2 -> 4 -> 3 -> 0

Safe sequence 13: 1 -> 3 -> 0 -> 2 -> 4  
Safe sequence 14: 1 -> 3 -> 0 -> 4 -> 2  
Safe sequence 15: 1 -> 3 -> 2 -> 0 -> 4  
Safe sequence 16: 1 -> 3 -> 2 -> 4 -> 0  
Safe sequence 17: 1 -> 3 -> 4 -> 0 -> 2  
Safe sequence 18: 1 -> 3 -> 4 -> 2 -> 0  
Safe sequence 19: 1 -> 4 -> 0 -> 2 -> 3  
Safe sequence 20: 1 -> 4 -> 0 -> 3 -> 2  
Safe sequence 21: 1 -> 4 -> 2 -> 0 -> 3  
Safe sequence 22: 1 -> 4 -> 2 -> 3 -> 0  
Safe sequence 23: 1 -> 4 -> 3 -> 0 -> 2  
Safe sequence 24: 1 -> 4 -> 3 -> 2 -> 0  
Safe sequence 25: 3 -> 1 -> 0 -> 2 -> 4  
Safe sequence 26: 3 -> 1 -> 0 -> 4 -> 2  
Safe sequence 27: 3 -> 1 -> 2 -> 0 -> 4  
Safe sequence 28: 3 -> 1 -> 2 -> 4 -> 0  
Safe sequence 29: 3 -> 1 -> 4 -> 0 -> 2  
Safe sequence 30: 3 -> 1 -> 4 -> 2 -> 0  
Safe sequence 31: 3 -> 4 -> 1 -> 0 -> 2  
Safe sequence 32: 3 -> 4 -> 1 -> 2 -> 0

---

END

---