**PROGRAM-6**

**Aim:** Implement Bully Election Algorithm

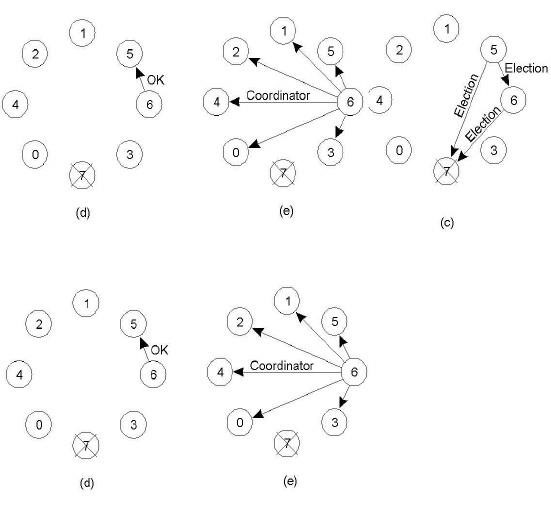
**Theory:**

**Election Algorithms**

Election algorithms choose a process from group of processors to act as a coordinator. If the coordinator process crashes due to some reasons, then a new coordinator is elected on other processor. Election algorithm basically determines where a new copy of coordinator should be restarted. Election algorithm assumes that every active process in the system has a unique priority number. The process with highest priority will be chosen as a new coordinator. Hence, when a coordinator fails, this algorithm elects that active process which has highest priority number. Then, this number is sent to every active process in the distributed system.

The Bully Election Process

1. P sends a message to the coordinator.
2. If coordinator does not respond to it within a time interval T, then it is assumed that coordinator has failed.
3. Now process P sends election message to every process with high priority number.
4. It waits for responses, if no one responds for time interval T then process P elects itself as a coordinator.
5. Then it sends a message to all lower priority number processes that it is elected as their new coordinator.
6. However, if an answer is received within time T from any other process Q,
   1. Process P again waits for time interval T’ to receive another message from Q that it has been elected as coordinator.
   2. If Q doesn’t respond within time interval T’ then it is assumed to have failed and algorithm is restarted.



Disadvantages:

* A large number of messages are sent, this can overload the system.
* There may be cases in very large systems that multiple coordinators get elected.

**Code:**

import java.io.\*;

import java.util.Scanner;

class Bully {

static int n;

static int pro[] = new int[100];

static int sta[] = new int[100];

static int co;

public static void main(String args[]) throws IOException {

System.out.print("Enter the number of process : ");

Scanner in = new Scanner(System.in);

n = in.nextInt();

int i, j, k, l, m;

for (i = 0; i < n; i++) {

System.out.println("For process " + (i + 1) + ":");

System.out.print("Status : ");

sta[i] = in.nextInt();

System.out.print("Priority : ");

pro[i] = in.nextInt();

}

System.out.println("Which process will initiate election?");

int ele = in.nextInt();

elect(ele);

System.out.println("Final coordinator is " + co);

}

static void elect(int ele) {

ele = ele - 1;

co = ele + 1;

for (int i = 0; i < n; i++) {

if (pro[ele] < pro[i]) {

System.out.println("Election message is sent from " + (ele + 1) + " to " + (i + 1));

if (sta[i] == 1)

elect(i + 1);

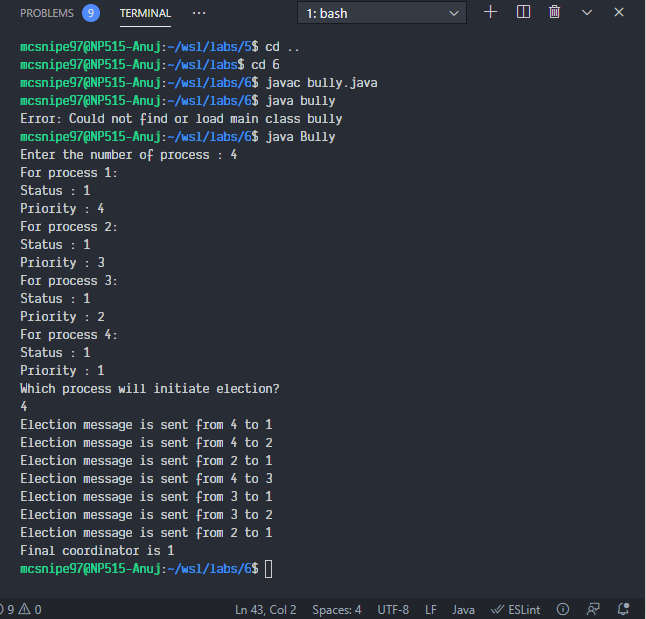
}

}

}

}

**Output:**



**Conclusion:**

1. We successfully implemented Bully-Election Algorithm.