PRAJAKTA DEOKULE
3330
C22019221332
ASSIGNMENT: WATER JUG PROBLEM using BFS

```
import java.util.*;
public class Main {

	public static void main(String[] args) throws java.lang.Exception{
		Scanner sc= new Scanner(System.in);

		System.out.println("Enter the capacity of Jug 1");
		int jug1= sc.nextInt();

		System.out.println("Enter the capacity of Jug 2");
		int jug2= sc.nextInt();

		System.out.println("Enter the target capacity to fill the jug with");
		int goal1= sc.nextInt();


		System.out.println("Enter the target capacity to fill the jug with");
		int goal2= sc.nextInt();

		getPathIfPossible(jug1, jug2, goal1,goal2);
	 }

	public static void getPathIfPossible(int jug1,int jug2,int goal1,int goal2) {

		boolean visited[][]=new boolean[jug1+1][jug2+1];

		Queue<State> q=new LinkedList<State>();
		q.add(new State(0,0));


		while(q.isEmpty()==false) {

			State cur=q.poll();

			//skip if already visited and overflowing state
			if(cur.j1>jug1||cur.j2>jug2||visited[cur.j1][cur.j2]) {
				continue;
			}


			visited[cur.j1][cur.j2]=true;

			if(cur.j1==goal1 && cur.j2==goal2) {

			//reached
			   int n = cur.states.size();
			System.out.println("The path followed by states of jug1    and jug2 is:");

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

	System.out.println("("+cur.states.get(i).j1+","+cur.states.get(i).j2+")");

			}
			//exit the program
			return;
```

```
        }

        //if the target is not yet achieved, then we have 3 cases left

        // fill empty and transfer
                    //fill one jug and empty the other
                     q.add(new State(jug1, 0, cur.states));
                     q.add(new State(0, jug2, cur.states));

                     //fill one jug and let other remain untouched
                     q.add(new State(jug1, cur.j2, cur.states));
                     q.add(new State(cur.j1, jug2, cur.states));

                     //empty one jug and let other remain untouched
                     q.add(new State(0, cur.j2, cur.states));
                     q.add(new State(cur.j1, 0, cur.states));

                     //transfer water from one jug to another until
                     //one becomes empty or other becomes full

                     if(cur.j1+cur.j2<jug1 && cur.j2>0)
                            q.add(new State(cur.j1+cur.j2,0,cur.states));

                     if(cur.j1+cur.j2<jug2 && cur.j1>0)
                            q.add(new State(0,cur.j1+cur.j2,cur.states));

                     if(cur.j1+cur.j2>=jug1 && cur.j2 >0)
                     {
                         q.add(new State(jug1,cur.j2-(jug1-cur.j1),cur.states));
                     }
                     if(cur.j1+cur.j2>=jug2 && cur.j1 >0)
                     {
                         q.add(new State(cur.j1-(jug2-cur.j2),jug2,cur.states));
                     }


              }

              //no possible solution
              System.out.println("Impossible to achieve target.");
        }
}
Enter the capacity of Jug 1
5
Enter the capacity of Jug 2
3
Enter the target capacity to fill the jug with
4
Enter the target capacity to fill the jug with
0
The path followed by states of jug1    and jug2 is:
(5,0)
(2,3)
(2,0)
(0,2)
(5,2)
(4,3)
(4,0)
```