

John Anderson 0893

CS 149 - 3

HW# 6

```
1) import java.util.*;
public class BoundedStack
{
    private Stack<Object> stack = new Stack<Object>();
    private int maxsize;
    public BoundedStack(int max)
    {
        maxsize = max;
    }
    public synchronized void push(Object e)
    {
        if(stack.size() < maxsize)
        {
            stack.push(e);
            this.notifyAll();
        }
    }
    public synchronized Object pop()
    {
        while(stack.size() == 0)
            try { wait(); }
            catch (InterruptedException e) { ; }
        return stack.pop();
    }
    public int getSize()
    {
        return stack.size();
    }
}
```

2)

Initialize:

Left, right, turn

While(true){

    Left.signal()

    Turn.wait();

    Right.signal();

    Turn.wait();

}

Left

Left.wait()

let car though

left.signal();

if(left == 0 )Turn.signal();

Right:

right.wait()

    let car though

right.signal()

if(right == 0 )Turn.signal();

3)

a) No, because the critical section of the eating is not protected. All this does is it making sure that forks aren't being held at the same time, but not making sure the eating's don't stack up or forks being released, so it's only protecting part of the critical section.

b) I would improve because would be signaling that you are done with the critical section after the eating and unlocking of the chopsticks is complete

c) It would get worse if you move it before the 2 wait because then it would not be protecting the critical section at all.

4) When they make a request for the first chop stick don't accept the request if there're no philosopher with 2 chop sticks and if there is only 1 chop stick remaining, grant it to the person with 1 chopstick.

5) When they make a request for the first chop stick don't accept the request if there no philosopher with 3 chop sticks or if there is only 2 chop stick remaining. Also accept requests with 2 chop sticks before ones with 1 and 0.

(wouldn't this be right because it's the same thing, but you just have to change the amount that the rule covers)

6)

A) Need matrix

	A	B	C	D
P0	0	0	0	0
P1	0	7	5	0
P2	1	0	0	2
P3	0	0	2	0
P4	0	6	4	2

B) yes, there are several sequences that satisfy safe requirements.

C) Yes, and it would change the allocated for P1 (1 4 2 0)

And make the need for P1(0 3 3 0)