

## CSci 151, 27 Oct 2008

```
public static int fact(int n) {
    if(n == 0) {
        return 1;
    } else {
        return n * fact(n - 1);
    }
}
```

```
class Frame {
    public int state;
    public int n;

    public Frame(int n) {
        this.state = 0;
        this.n = n;
    }
}
```

```
public static int fact(int n) {
    Stack<Frame> stack;
    stack = new Stack<Frame>();
    stack.push(new Frame(n));
    int ret = 0;
    while(!stack.isEmpty()) {
        Frame s = stack.peek();
        if(s.n == 0) {
            ret = 1;
            stack.pop();
        } else {
            if(s.state == 0) {
                s.state = 1;
                stack.push(new Frame(s.n - 1));
            } else {
                ret = s.n * ret;
                stack.pop();
            }
        }
    }
    return ret;
}
```

```
public static int fib(int n) {
    if(n <= 1) {
        return n;
    } else {
        return fib(n - 1)
            + fib(n - 2);
    }
}
```

```
class Frame {
    public int state;
    public int n;
    public int lastRet;

    public Frame(int n) {
        this.state = 0;
        this.n = n;
    }
}
```

```
public static int fib(int n) {
    Stack<Frame> stack;
    stack = new Stack<Frame>();
    stack.push(new Frame(n));
    int ret = 0;
    while(!stack.isEmpty()) {
        Frame s = stack.peek();
        if(s.n <= 1) {
            ret = s.n;
            stack.pop();
        } else {
            if(s.state == 0) {
                s.state = 1;
                stack.push(new Frame(s.n - 1));
            } else if(s.state == 1) {
                s.lastRet = ret;
                s.state = 2;
                stack.push(new Frame(s.n - 2));
            } else {
                ret = s.lastRet + ret;
                stack.pop();
            }
        }
    }
    return ret;
}
```

```

public void anagrams(String from, String pre) {
    if(from.length() <= 1) {
        System.out.println(pre + from);
    } else {
        for(int i = 0; i < from.length(); i++) {
            String bef = from.substring(0, i);
            char at = from.charAt(i);
            String aft = from.substring(i + 1);
            anagrams(bef + aft, pre + at);
        }
    }
}

public class Frame {
    public int i;
    public String from;
    public String pre;

    public Frame(String f, String p) {
        this.i = 0;
        this.from = f;
        this.pre = p;
    }
}

public void anagrams(String from, String pre) {
    Stack<Frame> stack;
    stack = new Stack<Frame>();
    stack.push(new Frame(from, pre));
    while(!stack.isEmpty()) {
        Frame s = stack.peek();
        if(s.from.length() <= 1) {
            System.out.println(s.pre + s.from);
            stack.pop();
        } else if(s.i < s.from.length()) {
            String bef = s.from.substring(0, s.i);
            char at = s.from.charAt(s.i);
            String aft = s.from.substring(s.i + 1);
            stack.push(new Frame(bef + aft, s.pre + at));
            s.i++;
        } else {
            stack.pop();
        }
    }
}

```