Recursion

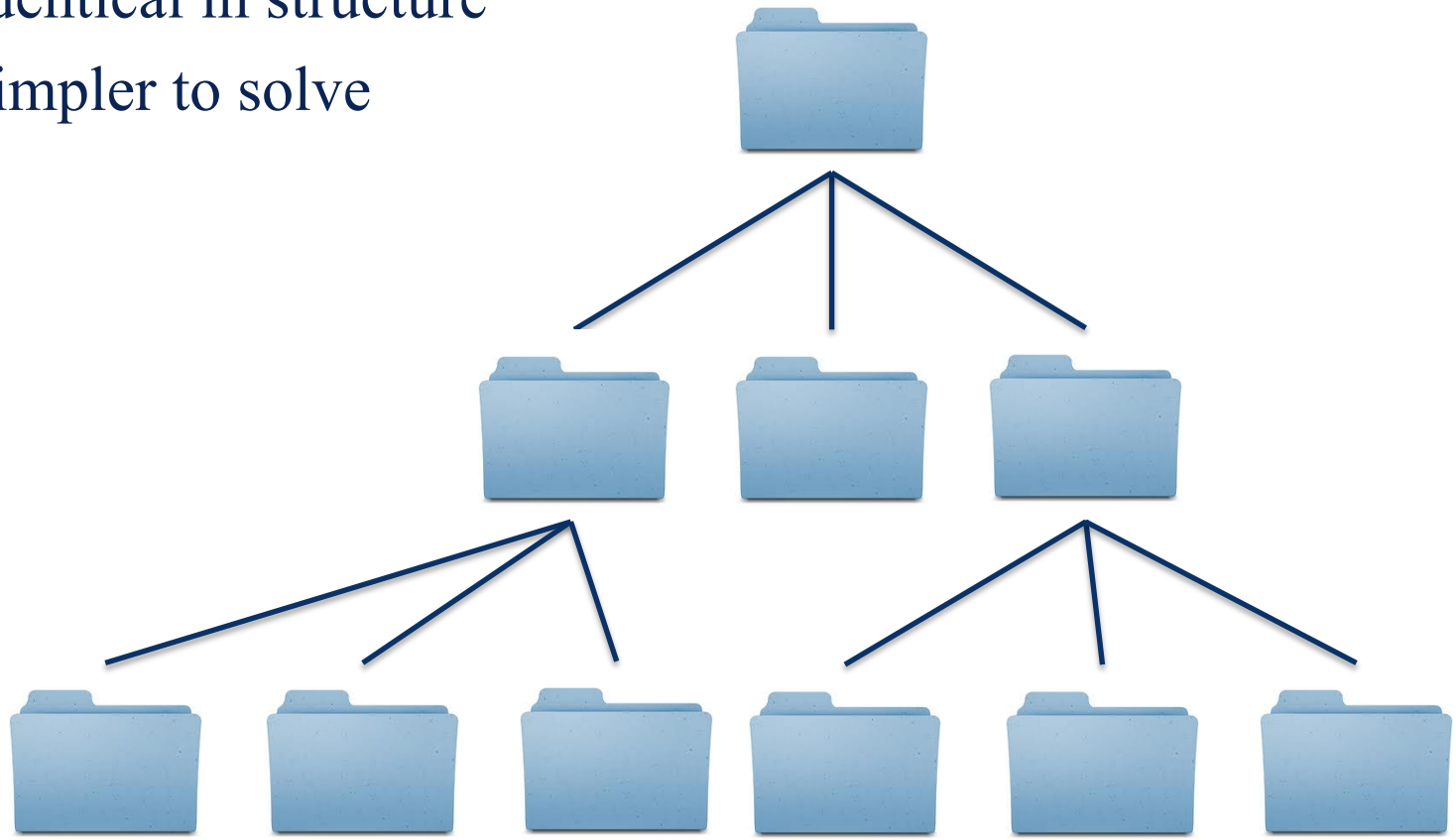
snarf today's code

Announcements

- Apt 4 – Due Oct 17
- Quiz Friday– Linked Lists
  - Can increase exam grade
- DNA – Due Oct 23

Last time

- Directory search
  - Reduce it to a sub problem
    1. identical in structure
    2. simpler to solve



General Structure

- void solve(ProblemClass instance){
  - if(instance *is simple*){ // base case
    - *solve instance directly*}

    - *divide into subinstance(s)*
    - solve(subinstance)
    - reassemble problem

Today

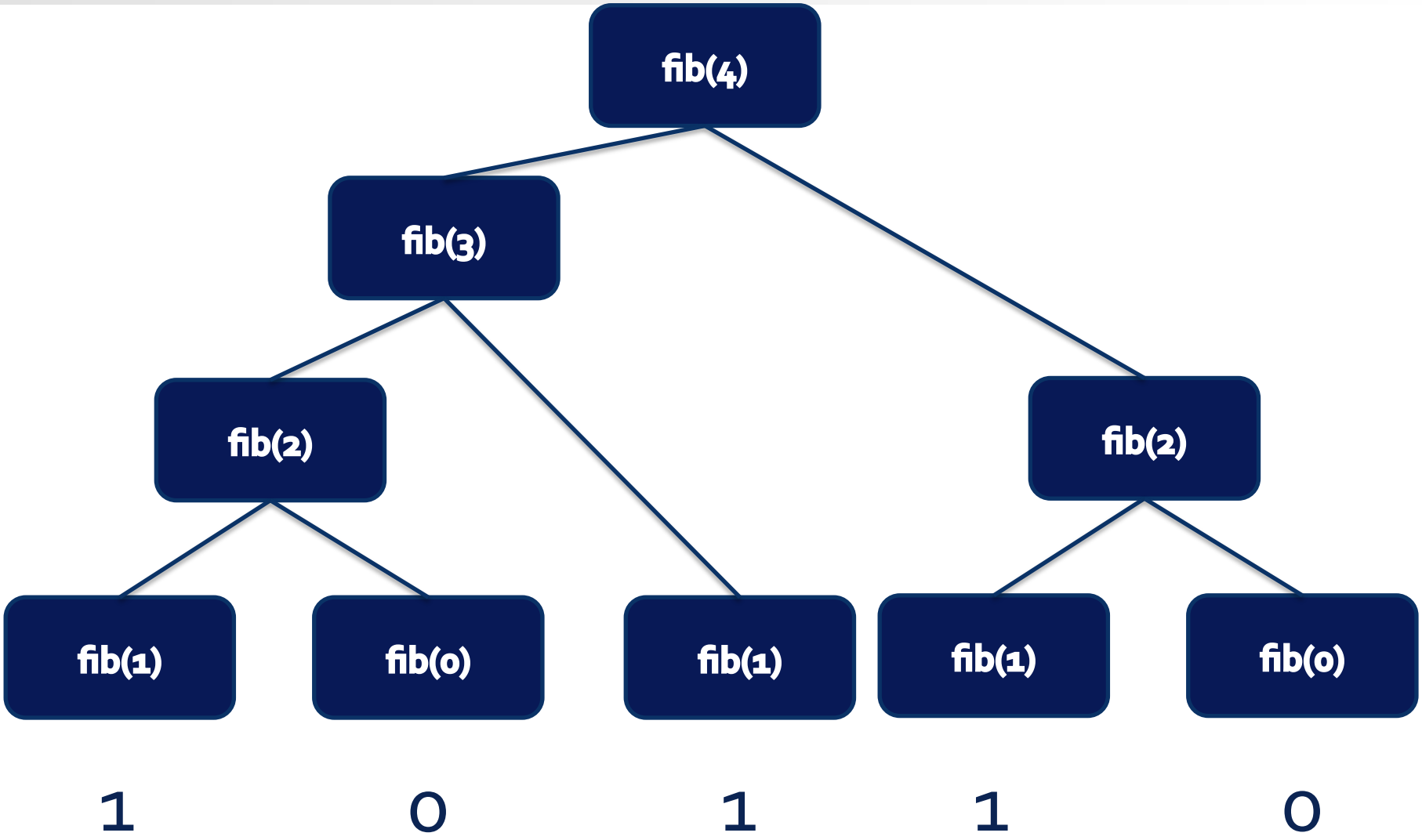
- Recursion and the call stack
- More examples:
  - Sorting
  - Flood fill

Fibonacci

- fib(0) = 0
- fib(1) = 1
- fib(2) = 1
- fib(3) = 2

- fib(n) = fib(n-1) + fib(n-2)

Fibonacci



Fibonacci

- When a function is called it is pushed to the call stack

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

fib(3);

Fibonacci

- When a function is called it is pushed to the call stack

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

fib(3);

Fibonacci

- Run code on top of stack

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

Fibonacci

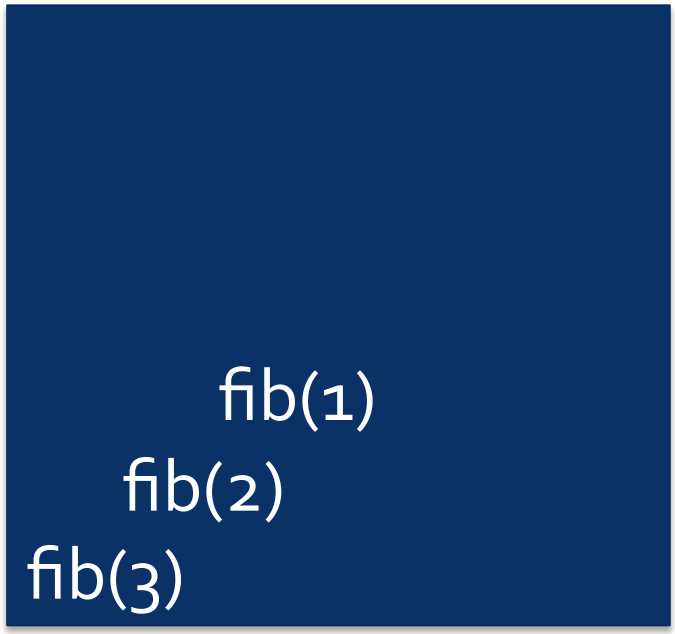
- Run code on top of stack

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

Fibonacci

- Run code on top of stack

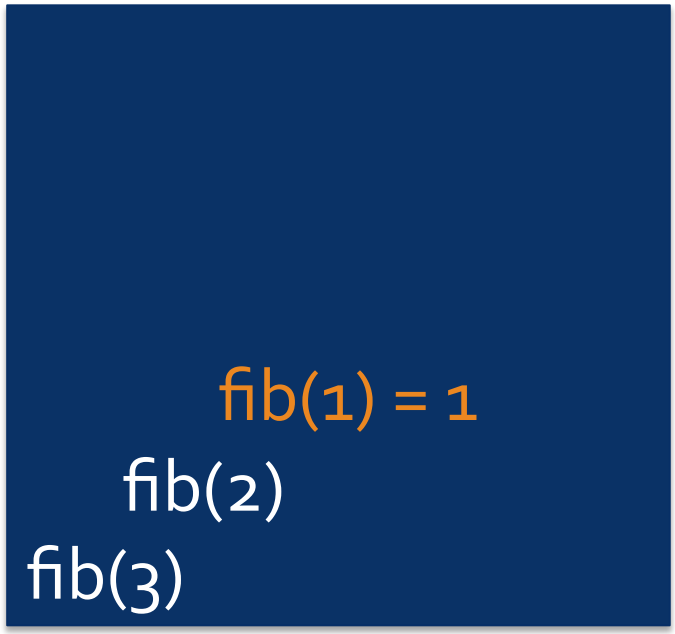
```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```



Fibonacci

- Pop when function completes

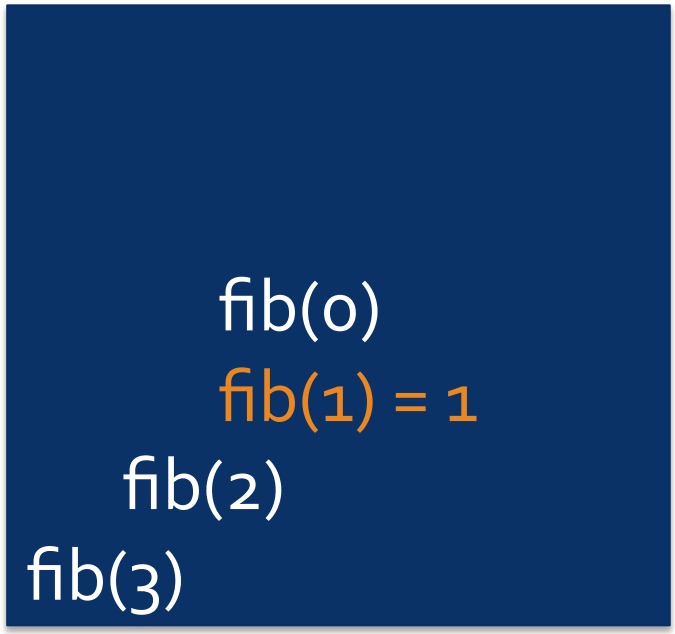
```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```



Fibonacci

- Continue from where you left off

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```



Fibonacci

- Continue from where you left off

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

fib(0) = 0
fib(1) = 1
fib(2)
fib(3)

Fibonacci

- Continue from where you left off

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

fib(0) = 0
fib(1) = 1
fib(2) = 1 + 0
fib(3)

Fibonacci

- Continue from where you left off

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

fib(1)
    fib(0) = 0
    fib(1) = 1
fib(2) = 1 + 0
fib(3)

Fibonacci

- Continue from where you left off

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

```
fib(1) = 1
    fib(0) = 0
    fib(1) = 1
fib(2) = 1 + 0
fib(3)
```

Fibonacci

- Continue from where you left off

```
int fib(int n){
   if(n==0) return 0;
   if(n==1) return 1;
   return fib(n-1) * fib(n-2);
}
```

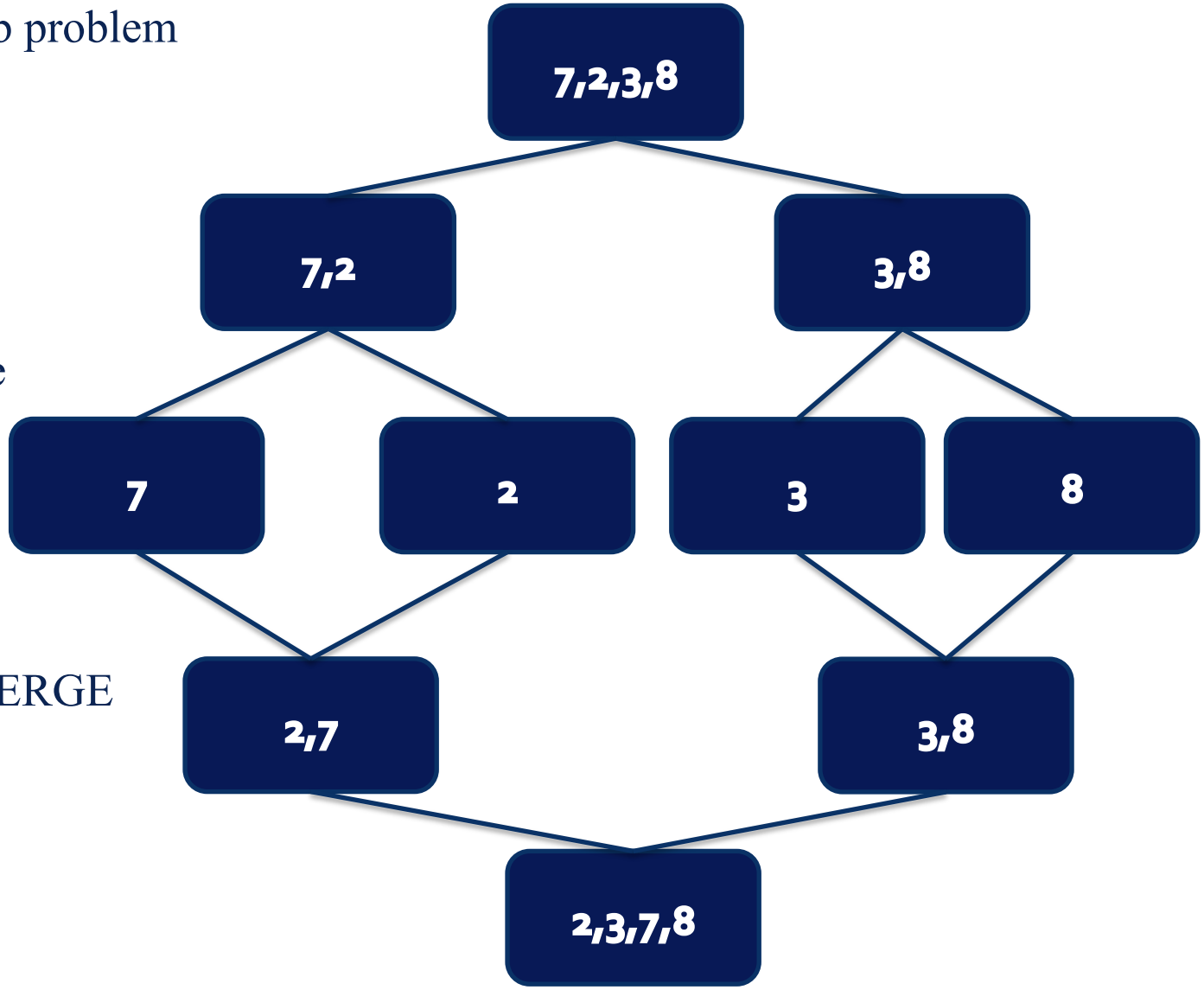
```
fib(1) = 1
    fib(0) = 0
    fib(1) = 1
fib(2) = 1 + 0
fib(3) = 1 + 1
```

Today

- Recursion and the call stack
- More examples:
  - Sorting
  - Flood Fill

MergeSort

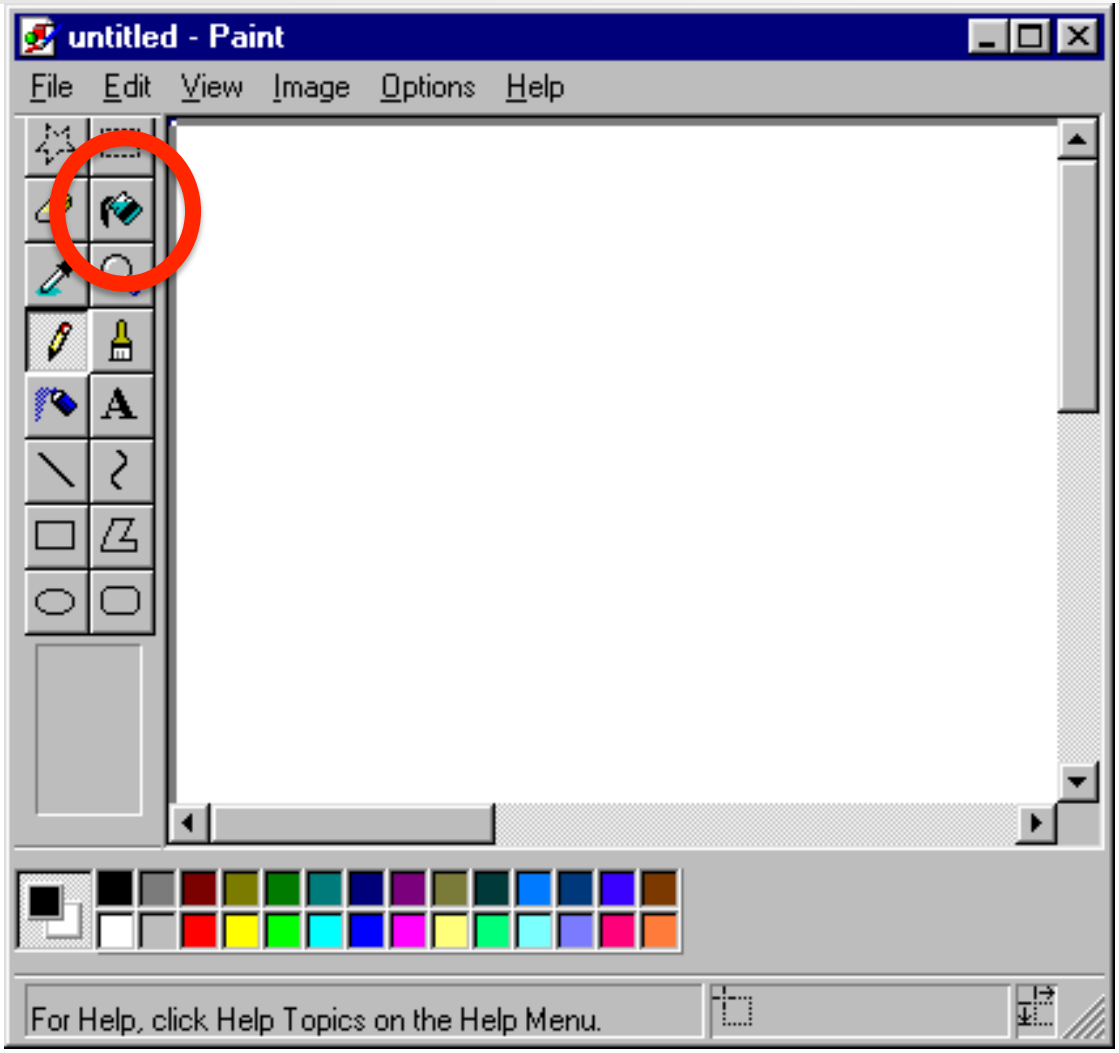
- Reduce to a sub problem
  - half the list
- Simple to solve
  - 1 element
- Once sorted MERGE lists together



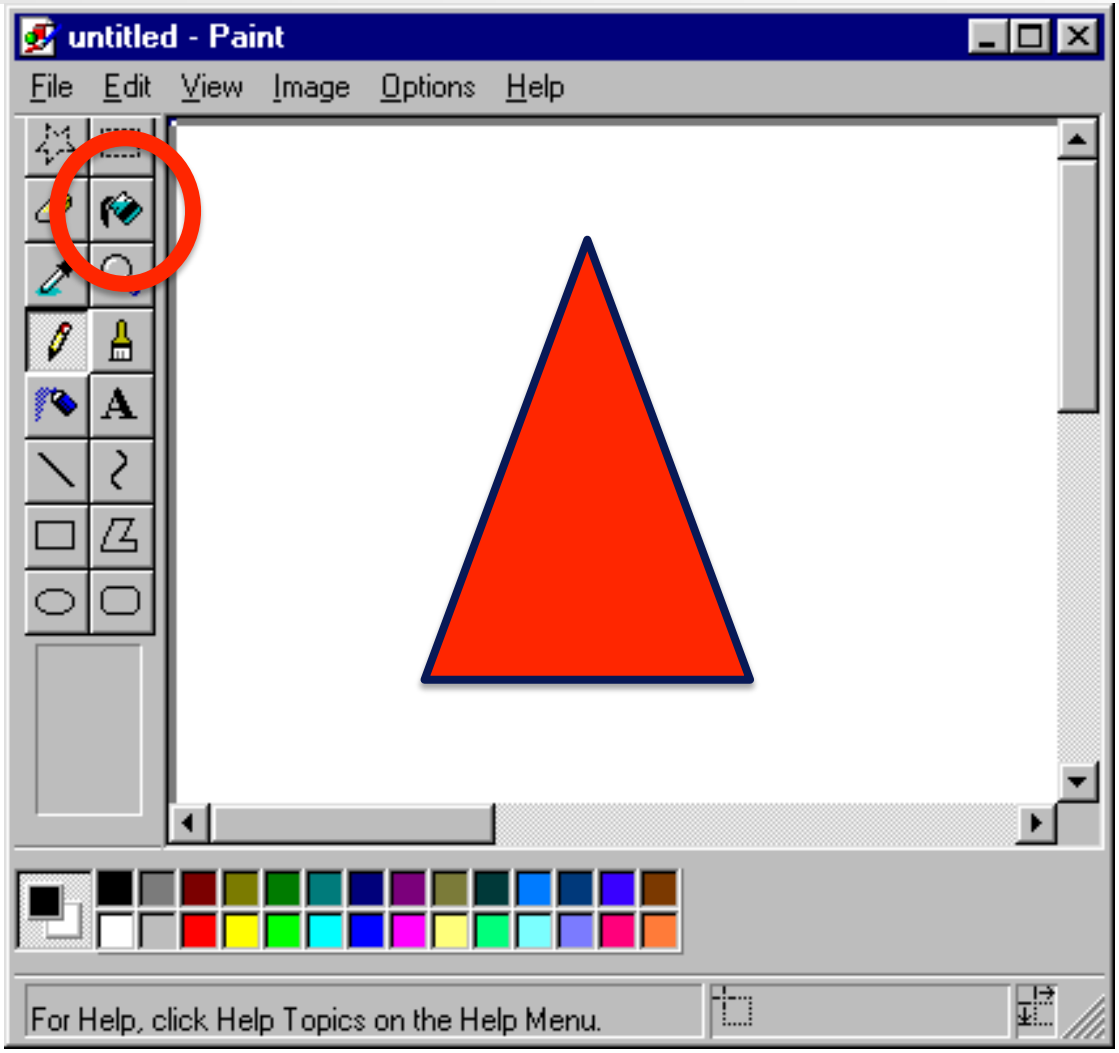
Today

- Recursion and the call stack
- More examples:
  - Sorting
  - Flood Fill

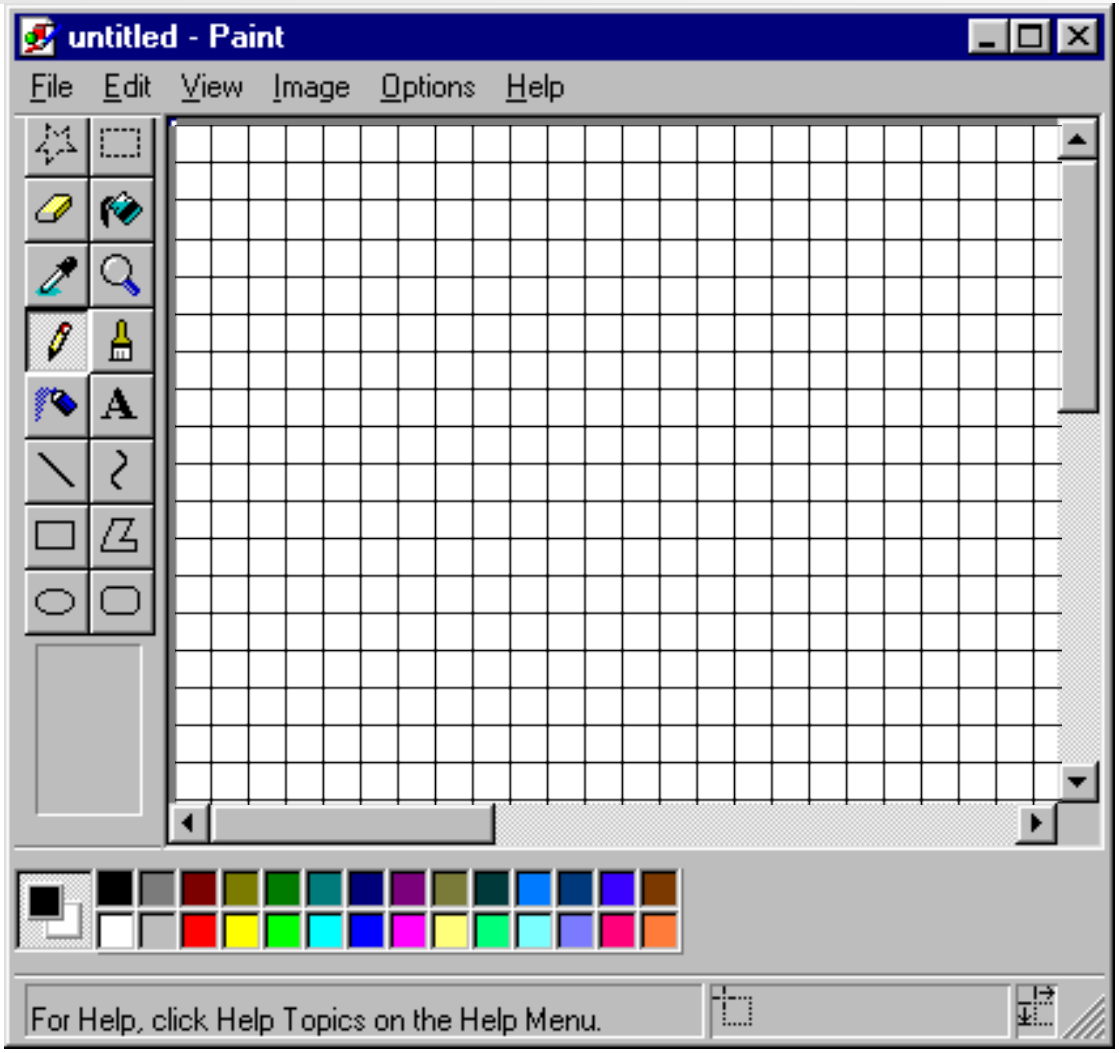
Flood Fill


untitled - Paint
File Edit View Image Options Help
For Help, click Help Topics on the Help Menu.

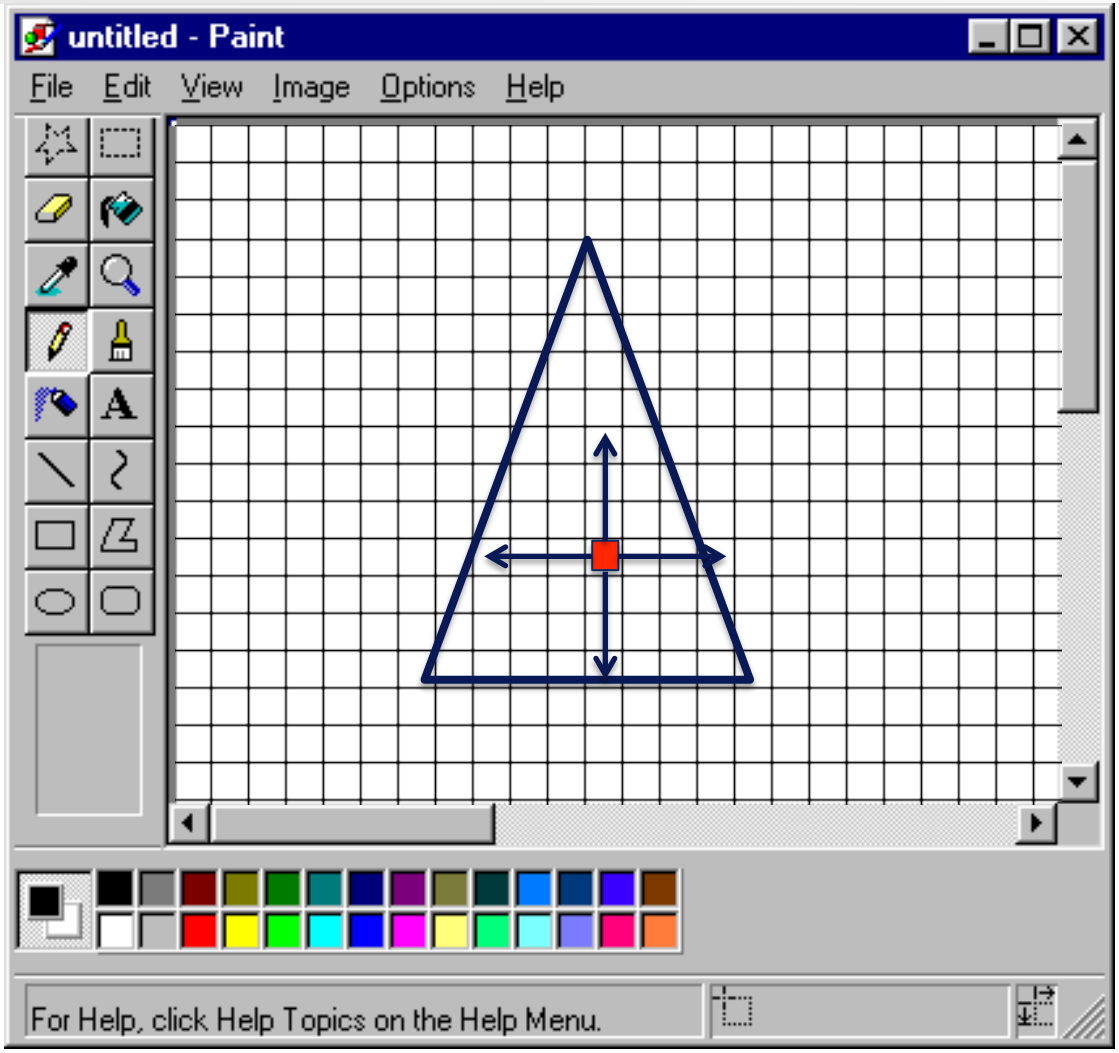
Flood Fill


untitled - Paint
File Edit View Image Options Help
For Help, click Help Topics on the Help Menu.

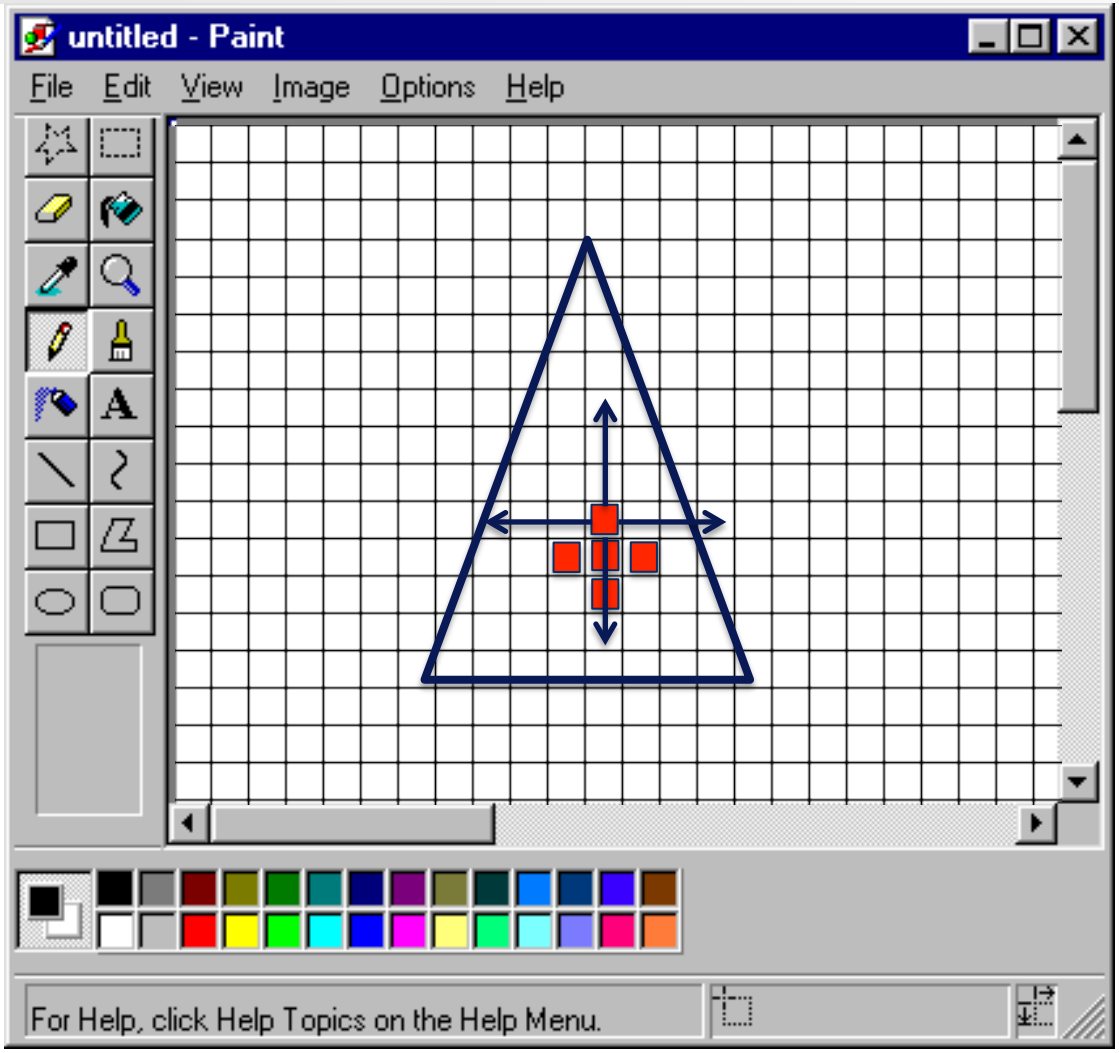
Flood Fill


untitled - Paint
File Edit View Image Options Help
For Help, click Help Topics on the Help Menu.

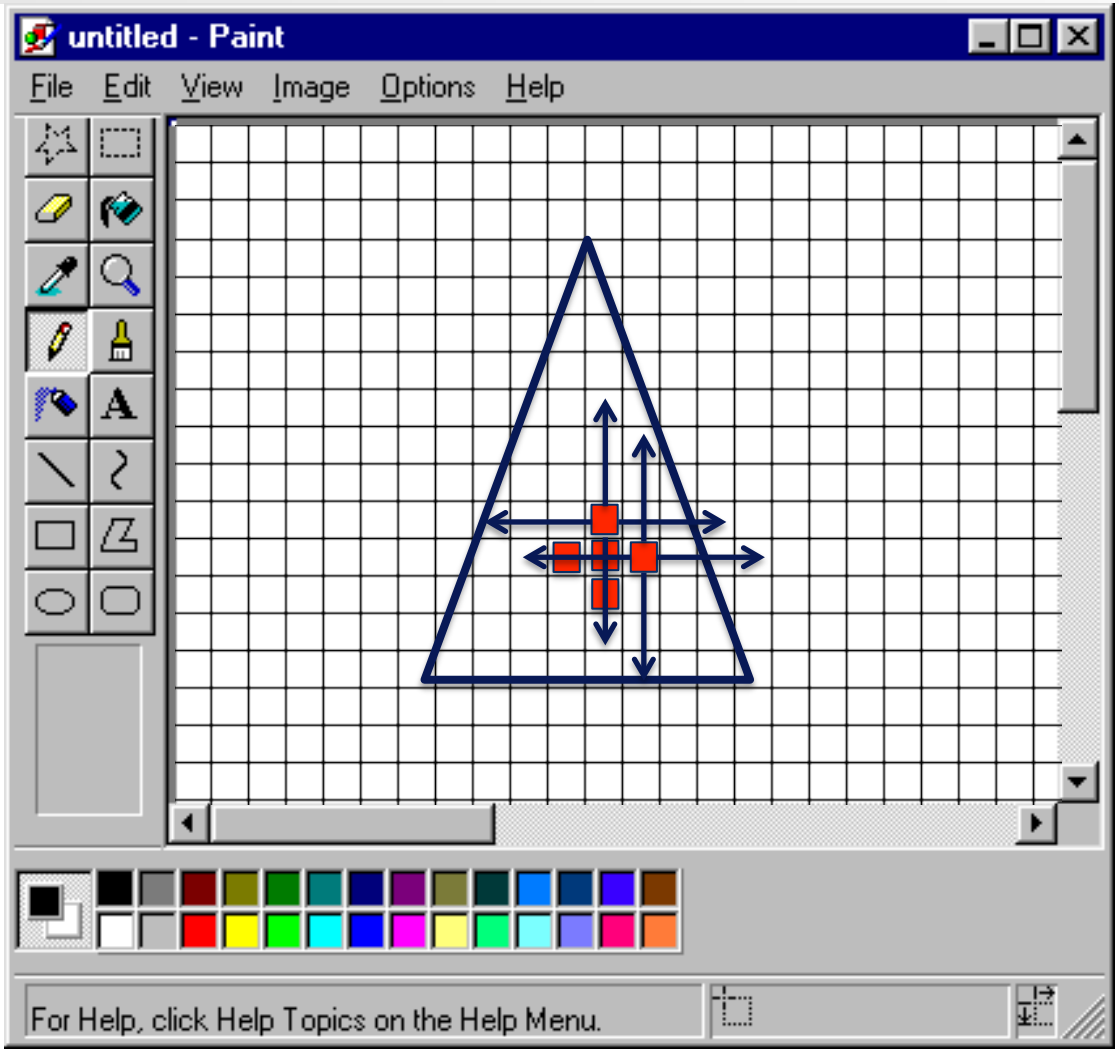
Flood Fill


untitled - Paint
File Edit View Image Options Help
For Help, click Help Topics on the Help Menu.

Flood Fill


untitled - Paint
File Edit View Image Options Help
For Help, click Help Topics on the Help Menu.

Flood Fill


untitled - Paint
File Edit View Image Options Help
For Help, click Help Topics on the Help Menu.

Today

- Recursion and the call stack
- More examples:
  - Sorting
  - Flood Fill