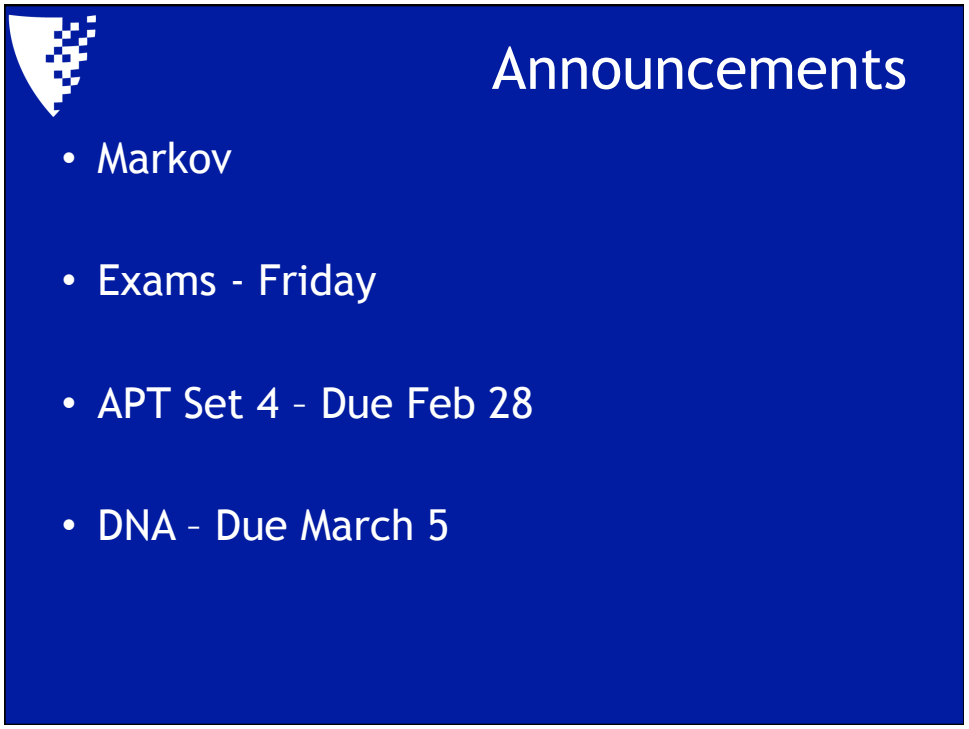
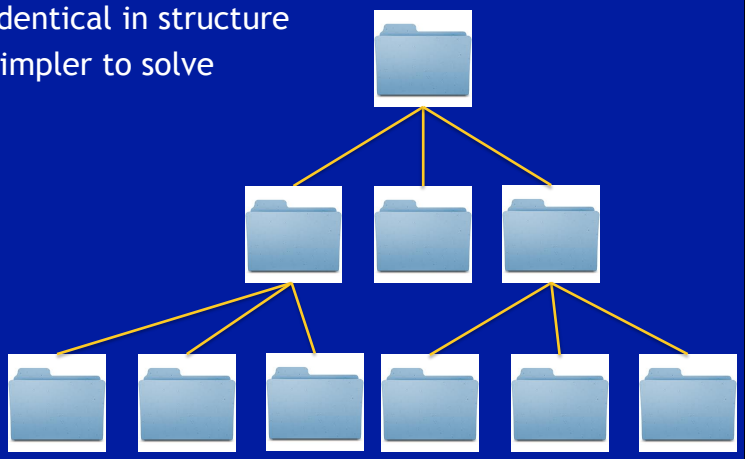

Recursion
snarf today's code


Announcements
• Markov
• Exams - Friday
• APT Set 4 – Due Feb 28
• DNA – Due March 5

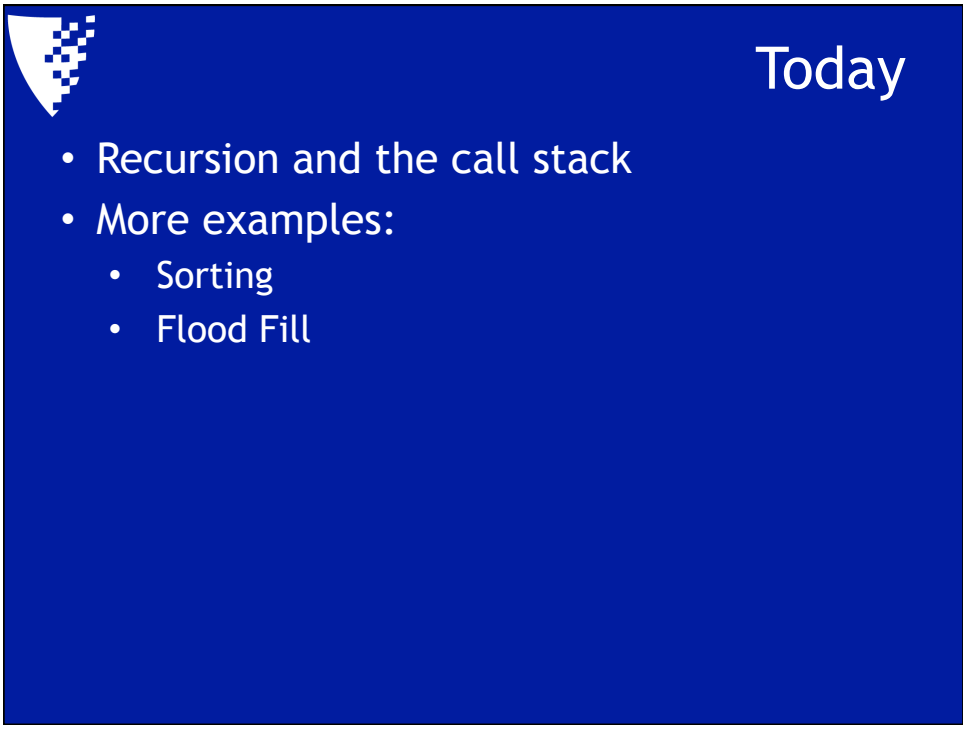
Last time

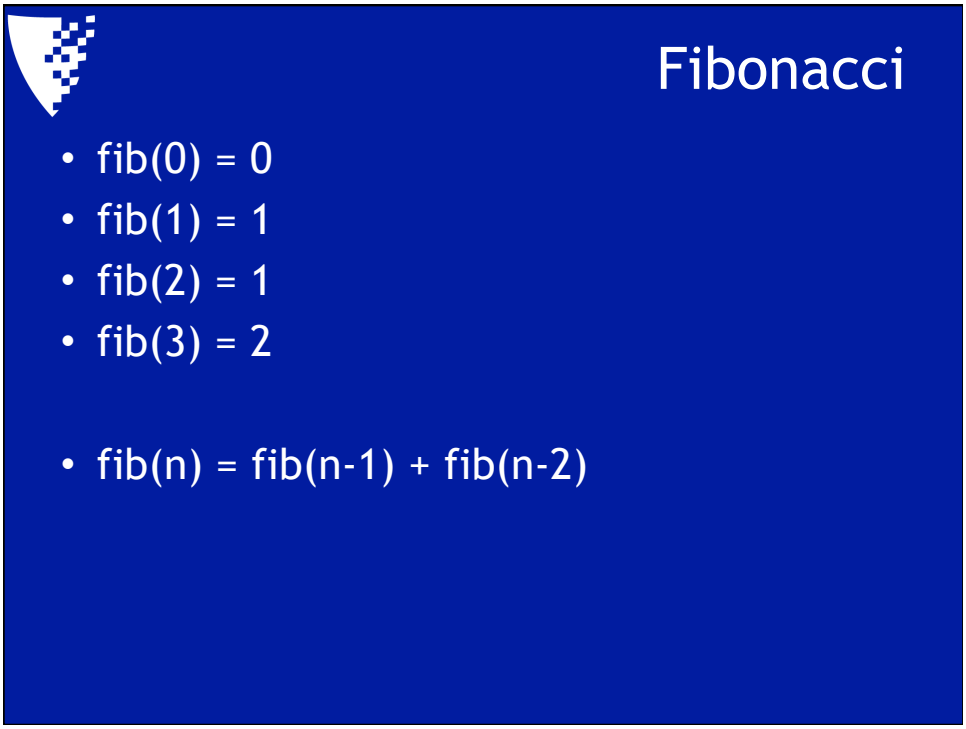
- Directory search
  - Reduce it to a sub problem
    1. identical in structure
    2. simpler to solve

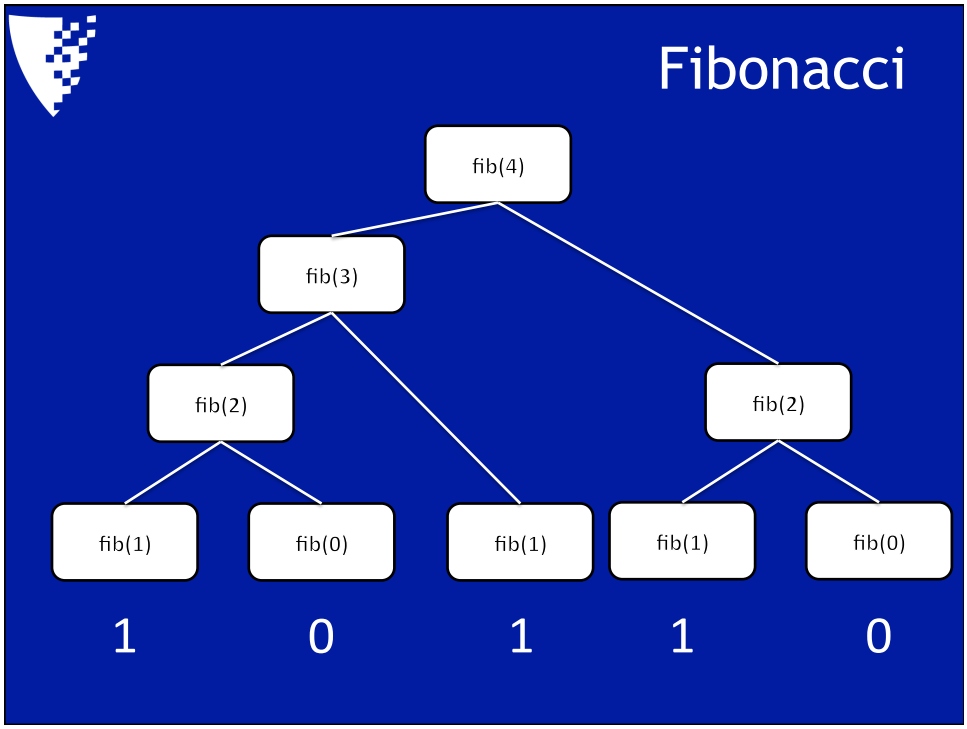


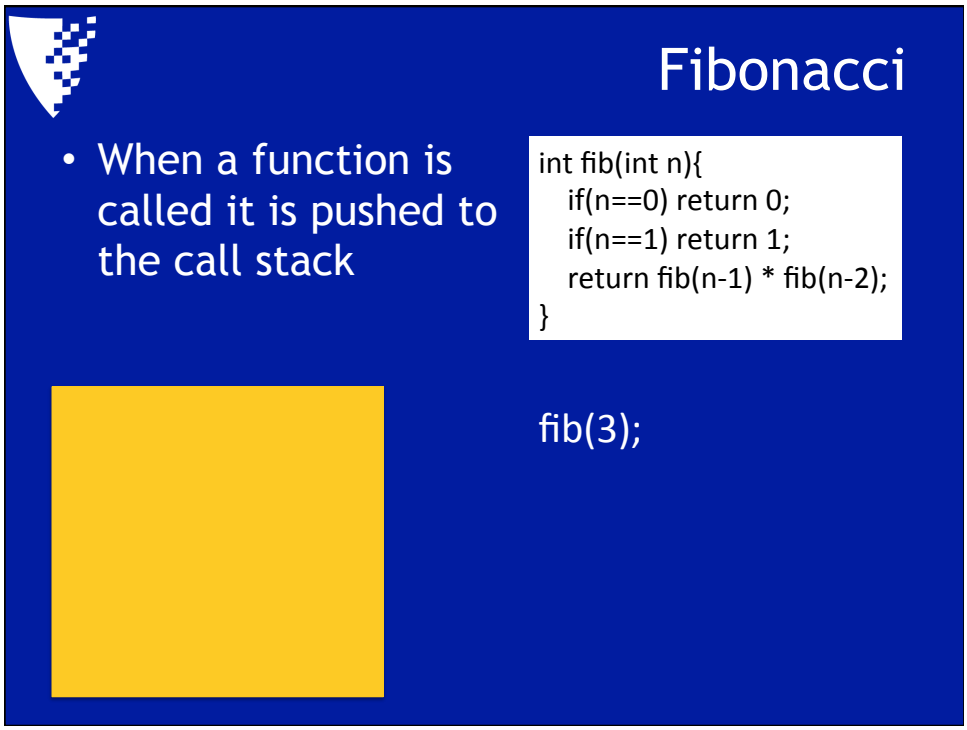
General Structure

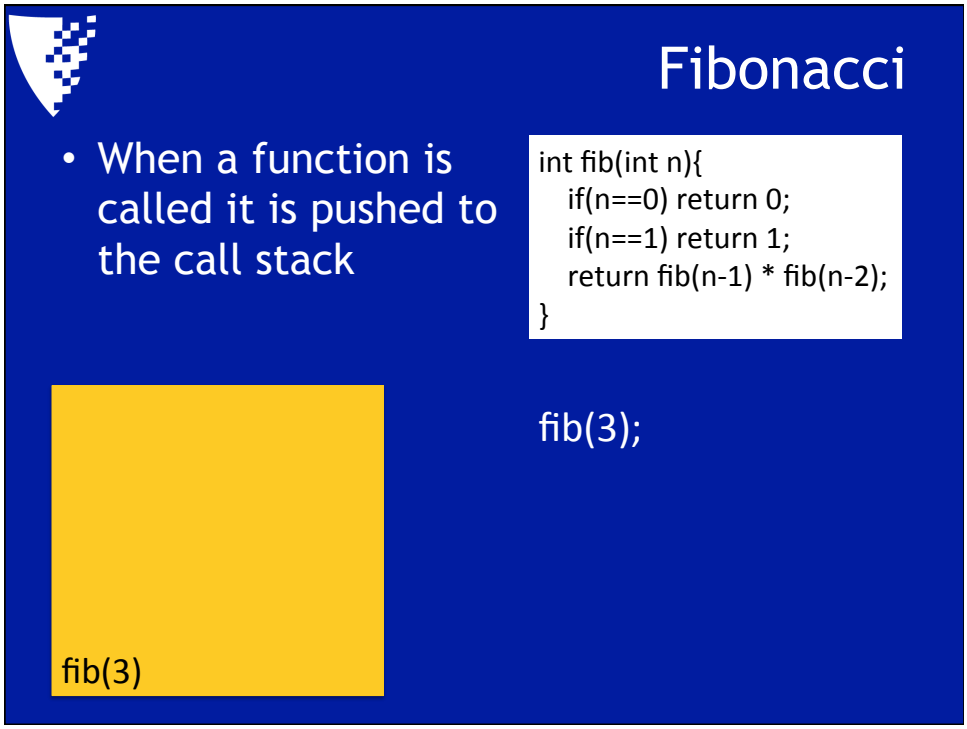
- void solve(ProblemClass instance){
  - if(instance *is simple*){
    - *solve instance directly*}
  - else{
    - *divide into subinstance(s)*
    - solve(subinstance)
    - reassemble problem
  - }
- }

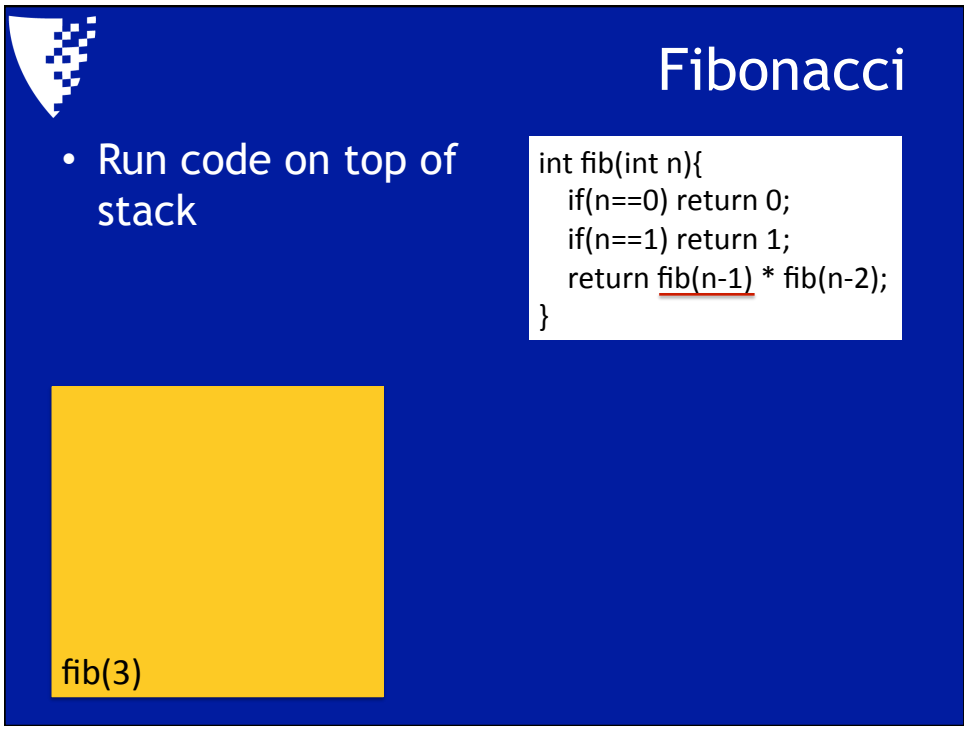

Today
• Recursion and the call stack
• More examples:
• Sorting
• Flood Fill

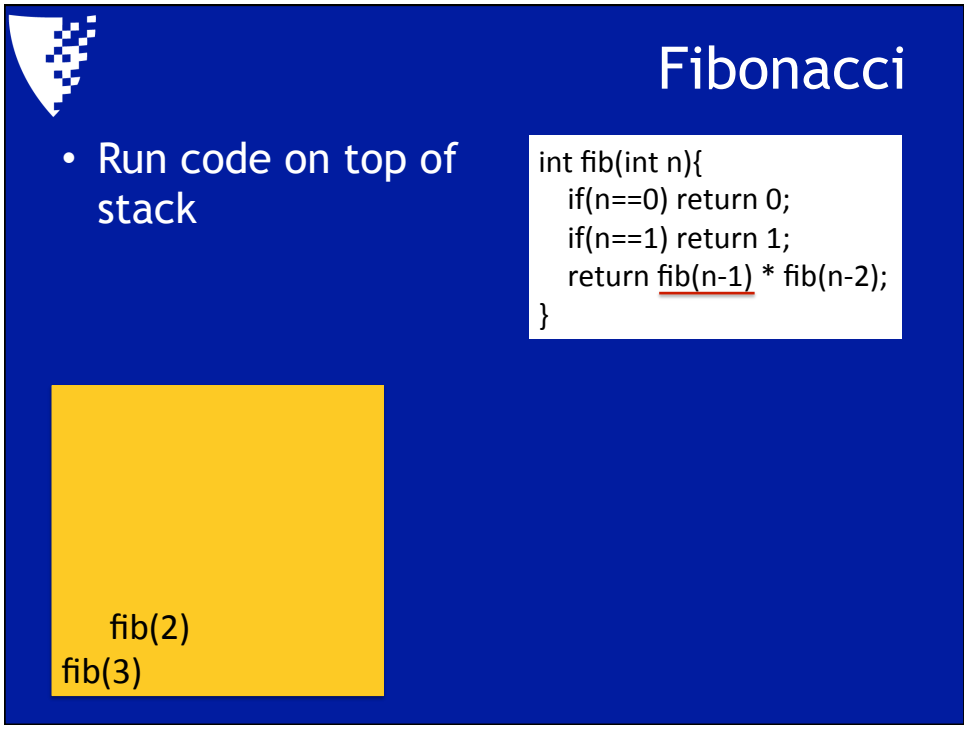

Fibonacci
• fib(0) = 0
• fib(1) = 1
• fib(2) = 1
• fib(3) = 2
• fib(n) = fib(n-1) + fib(n-2)

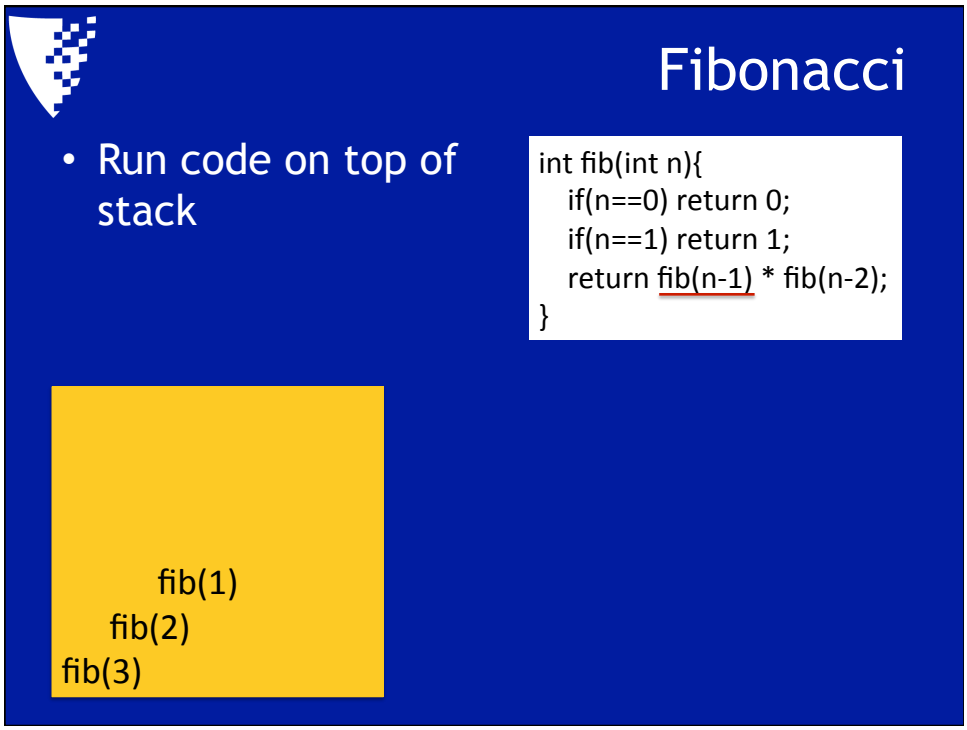

Fibonacci
fib(4)
fib(3)
fib(2)
fib(2)
fib(1)
fib(0)
fib(1)
fib(1)
fib(0)
1
0
1
1
0

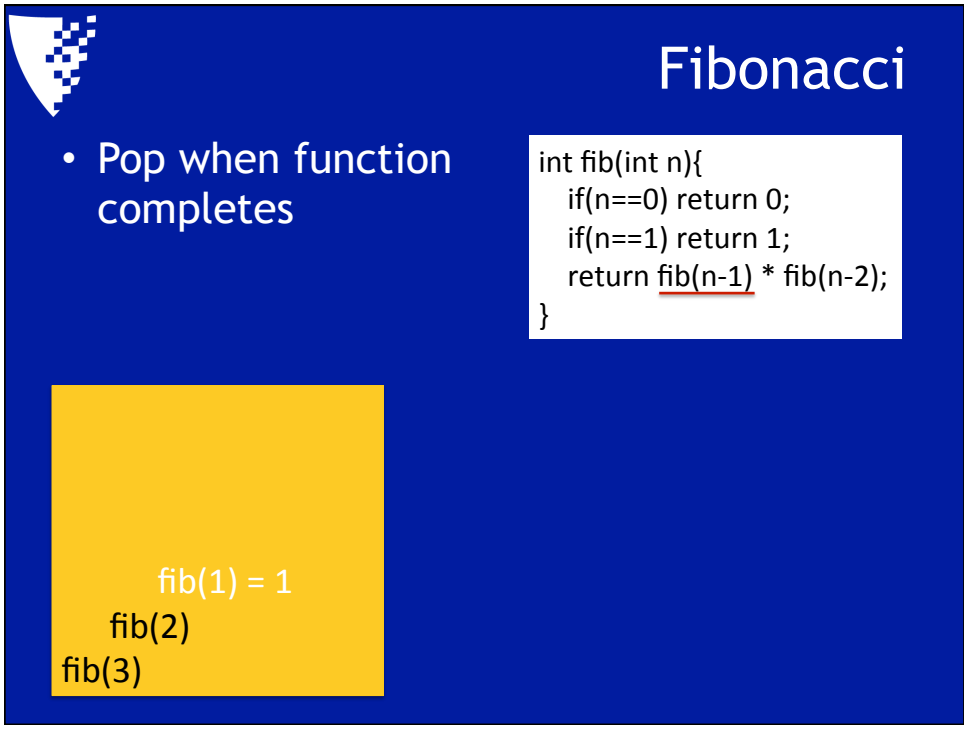

Fibonacci
• When a function is called it is pushed to the call stack
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(3);

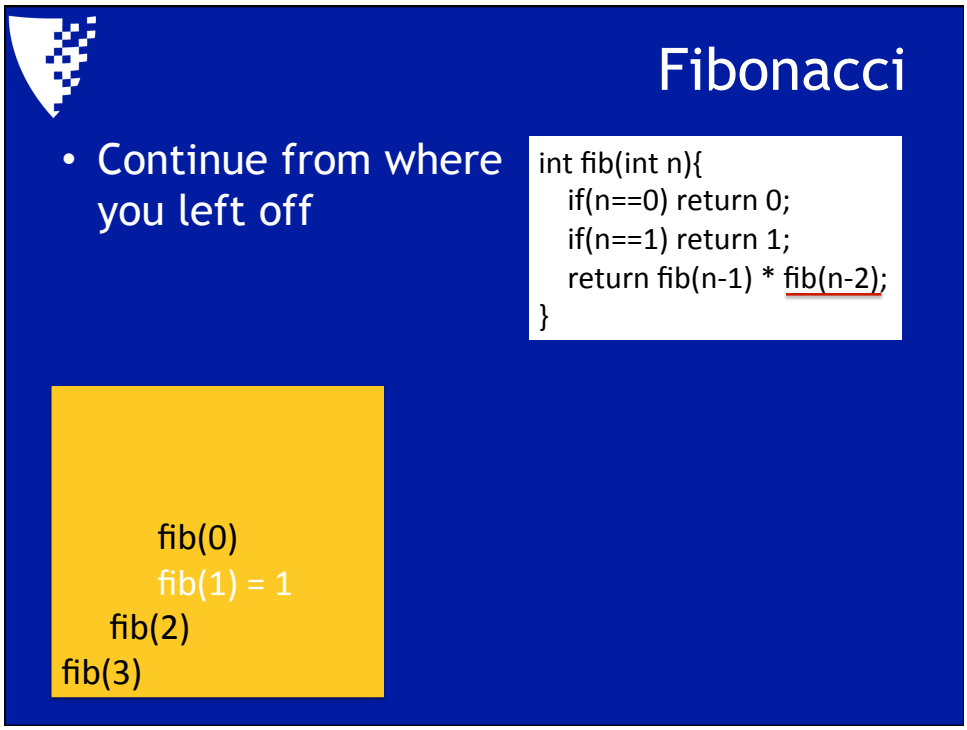

Fibonacci
• When a function is called it is pushed to the call stack
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(3);
fib(3)

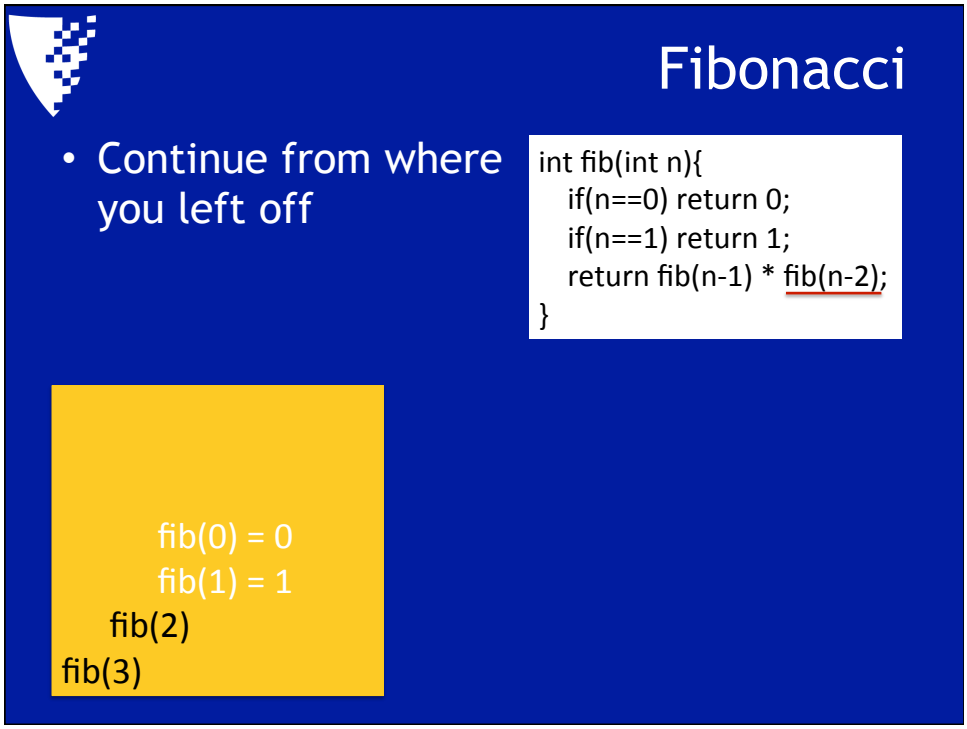

Fibonacci
• Run code on top of stack
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(3)

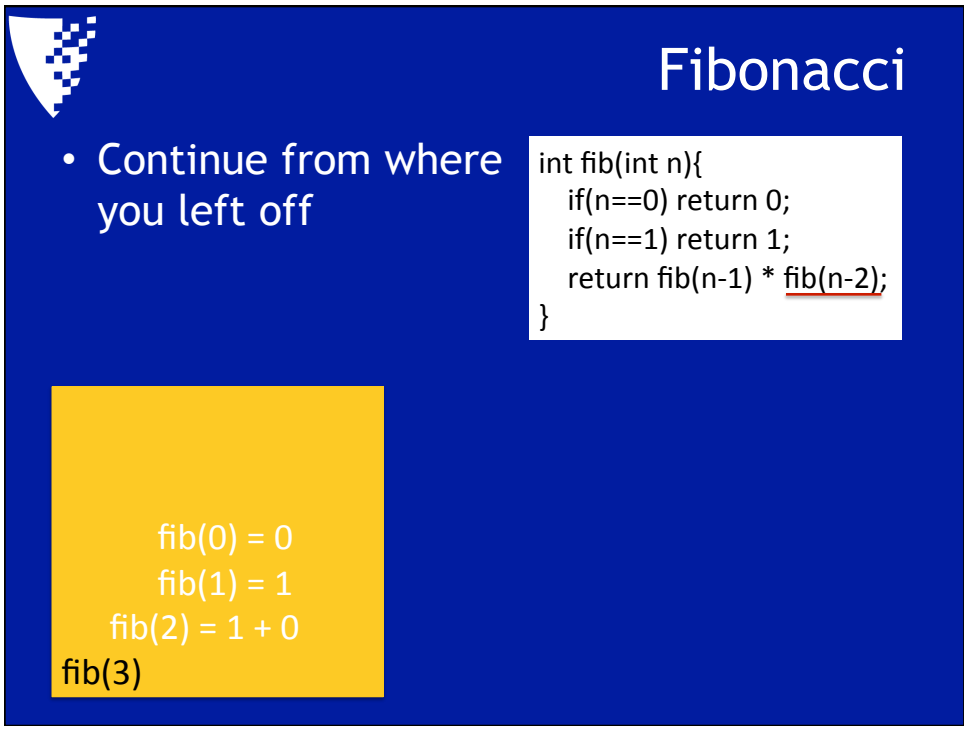

Fibonacci
• Run code on top of stack
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(2)
fib(3)

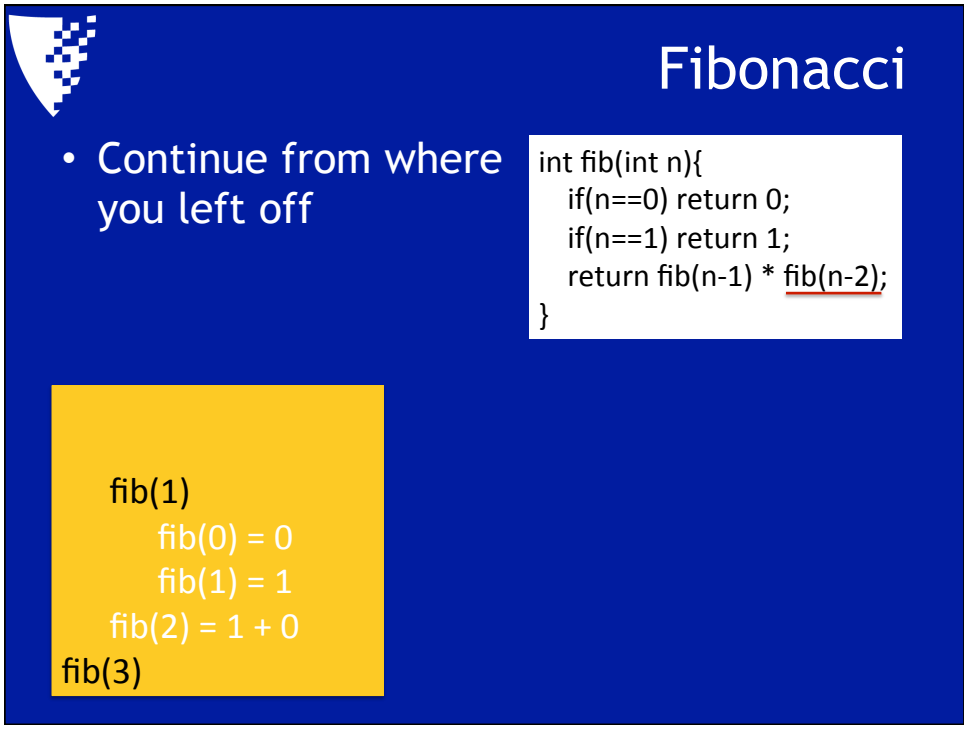

Fibonacci
• Run code on top of stack
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(1)
fib(2)
fib(3)

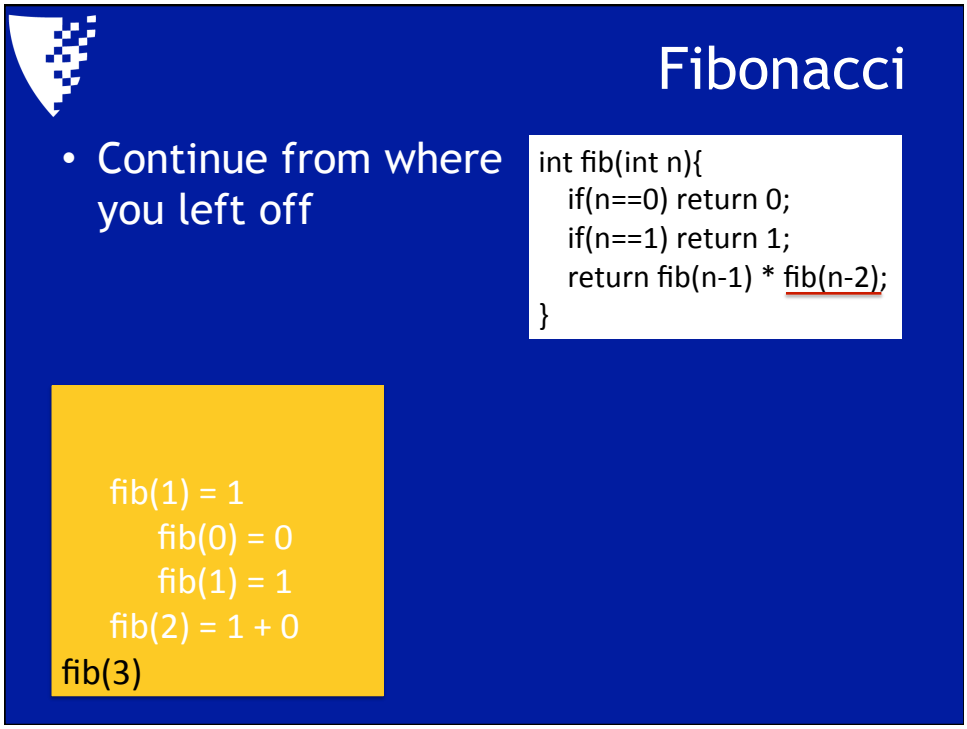

Fibonacci
• Pop when function completes
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(1) = 1
fib(2)
fib(3)

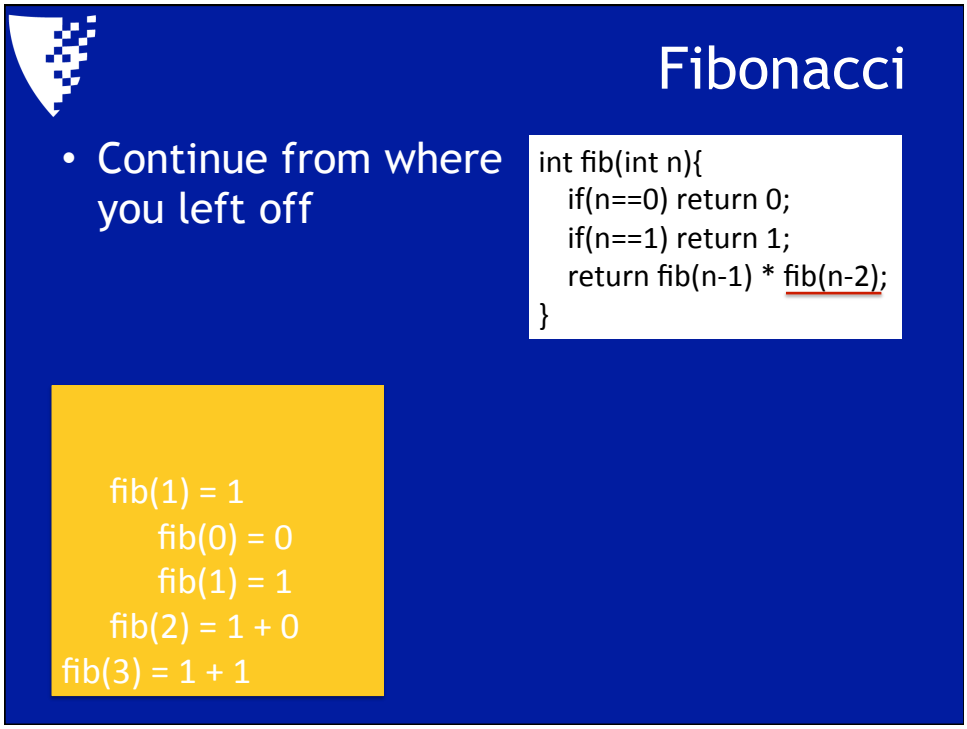

Fibonacci
• Continue from where you left off
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(0)
fib(1) = 1
fib(2)
fib(3)

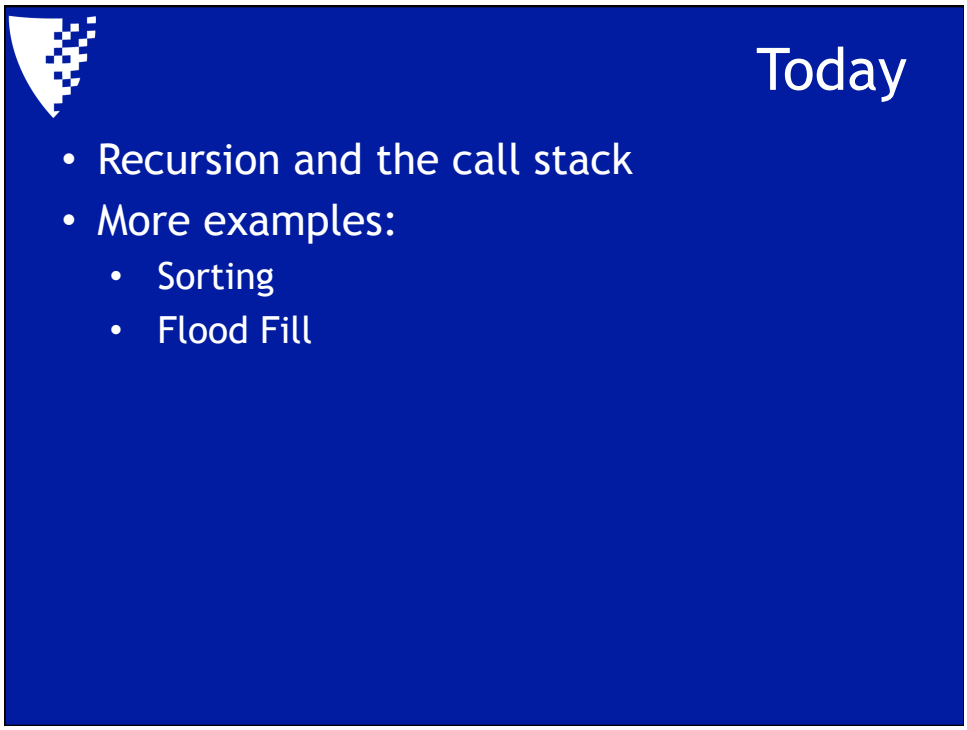

Fibonacci
• Continue from where you left off
int fib(int n){
if(n==0) return 0;
if(n==1) return 1;
return fib(n-1) * fib(n-2);
}
fib(0) = 0
fib(1) = 1
fib(2)
fib(3)


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

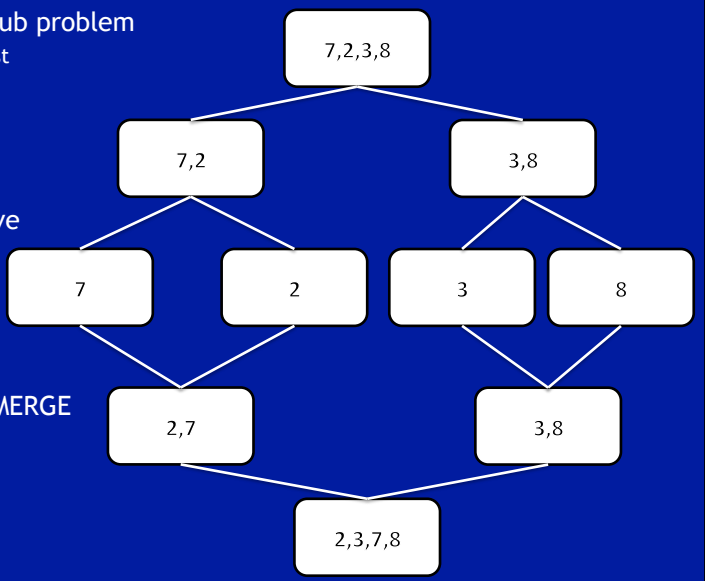

Today
• Recursion and the call stack
• More examples:
• Sorting
• Flood Fill

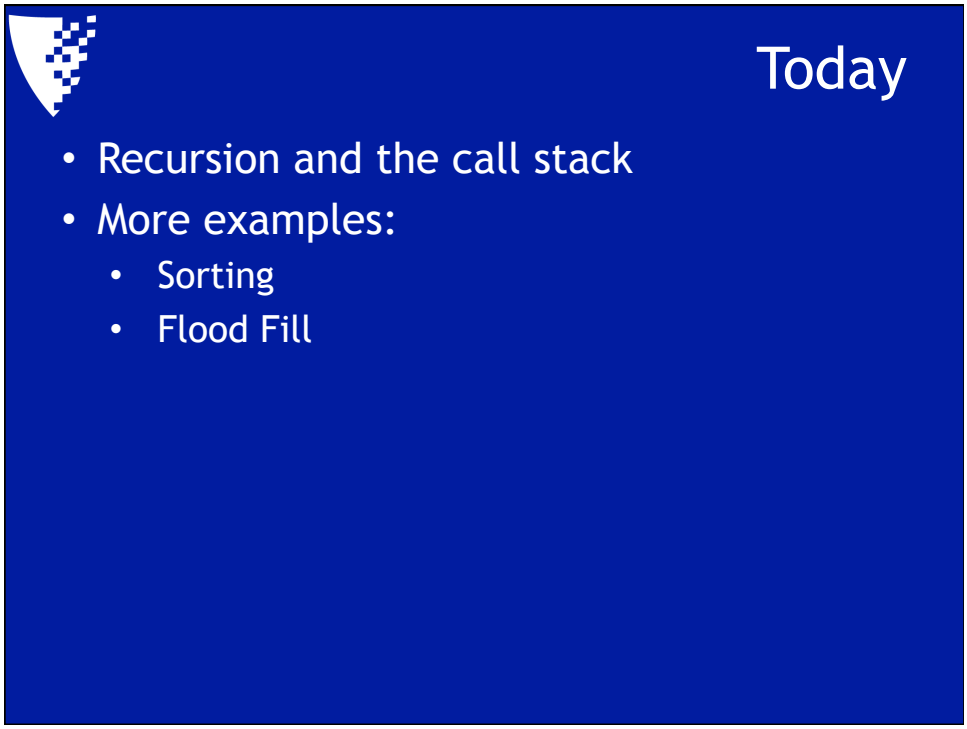
Merge Sort

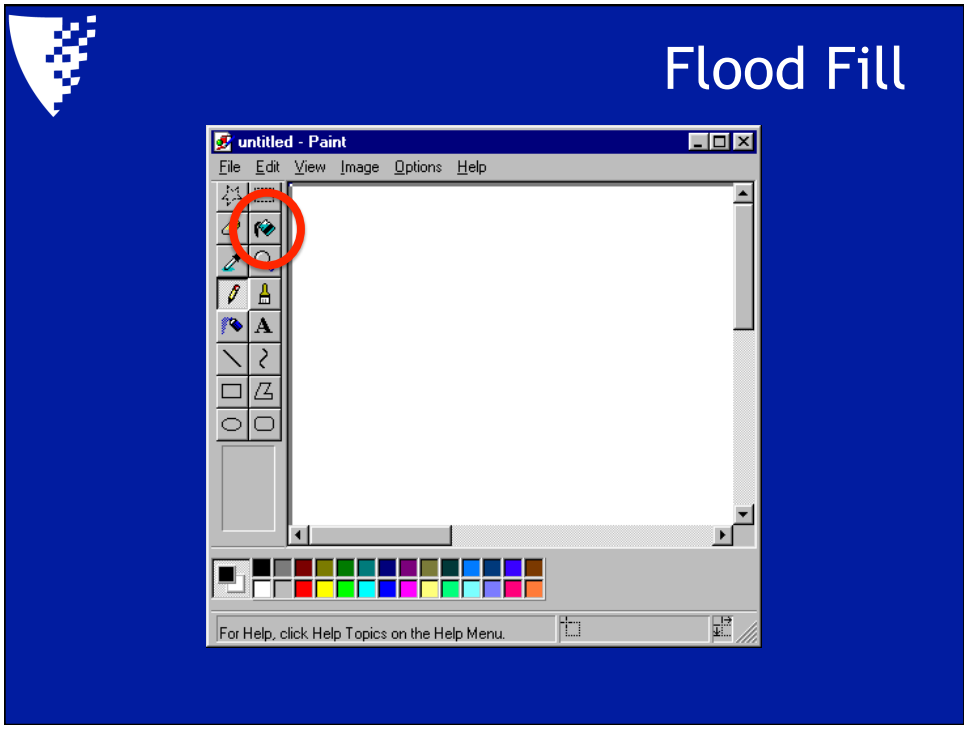
- Sort an array of integers into increasing order
- http://goo.gl/3hrUj

MergeSort

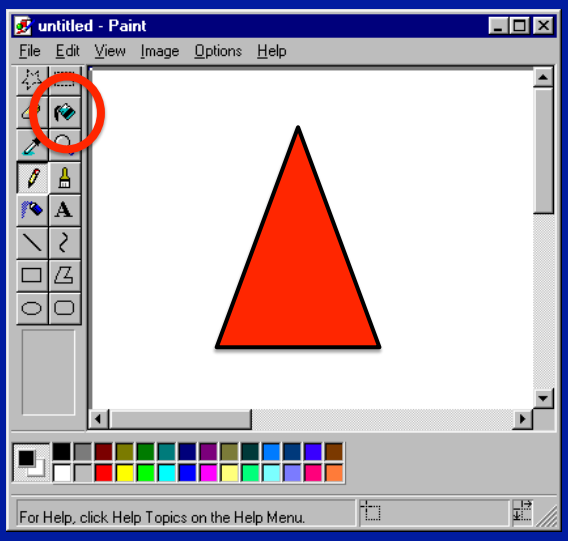
- Reduce to a sub problem
  - half the list
- Simple to solve
  - 1 element
- Once sorted MERGE lists together



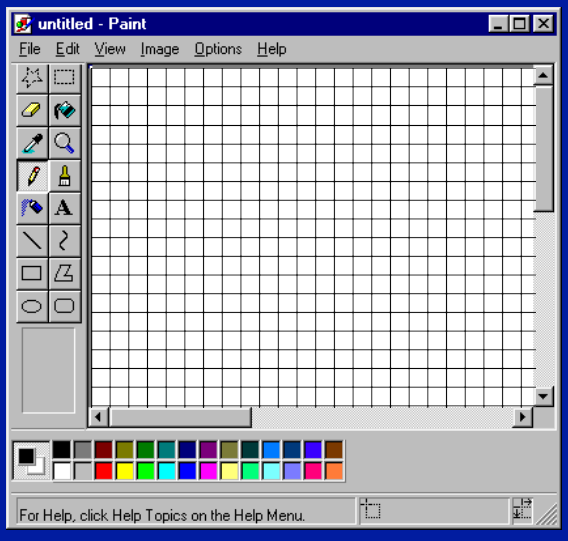

Today
• Recursion and the call stack
• More examples:
• Sorting
• Flood Fill


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

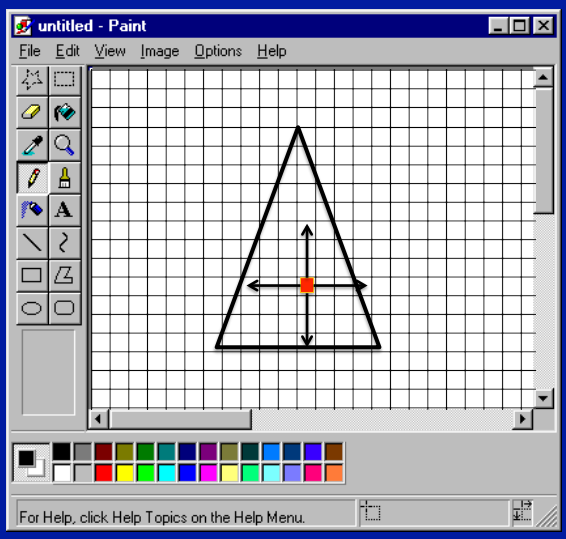
Flood Fill



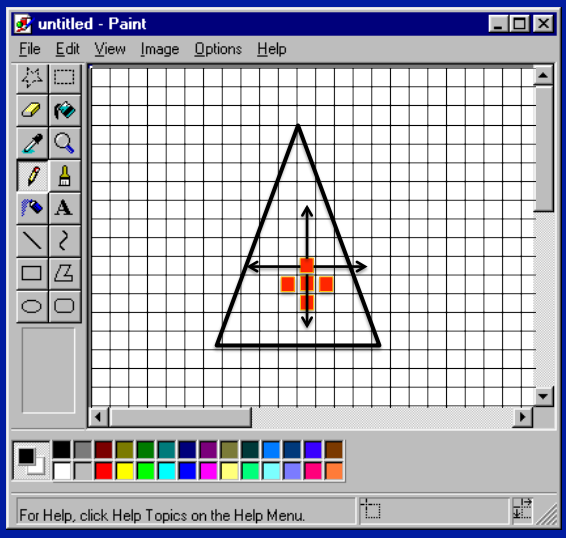
Flood Fill



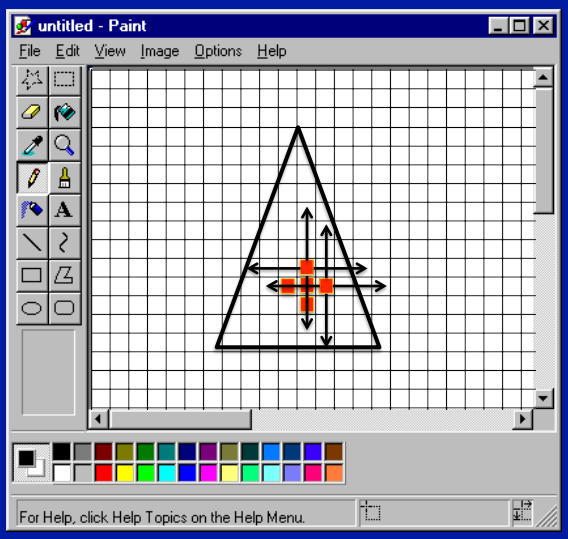
Flood Fill



Flood Fill



Flood Fill



Today

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