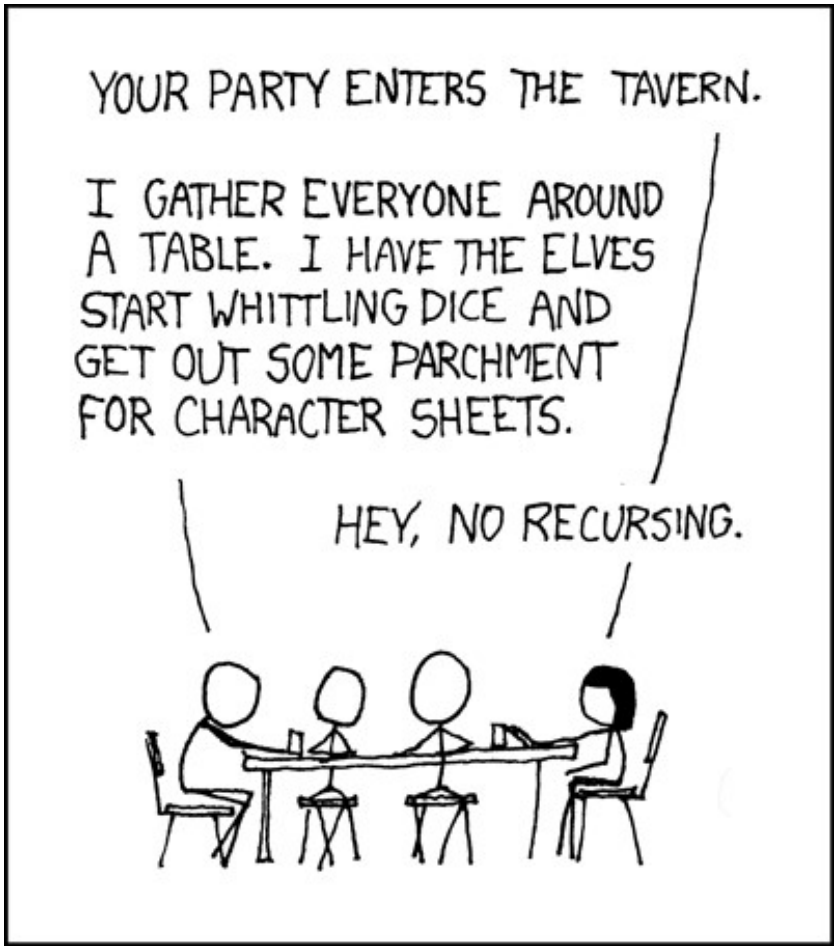
CS 161, Lecture 13: Recursion



What is Recursion?

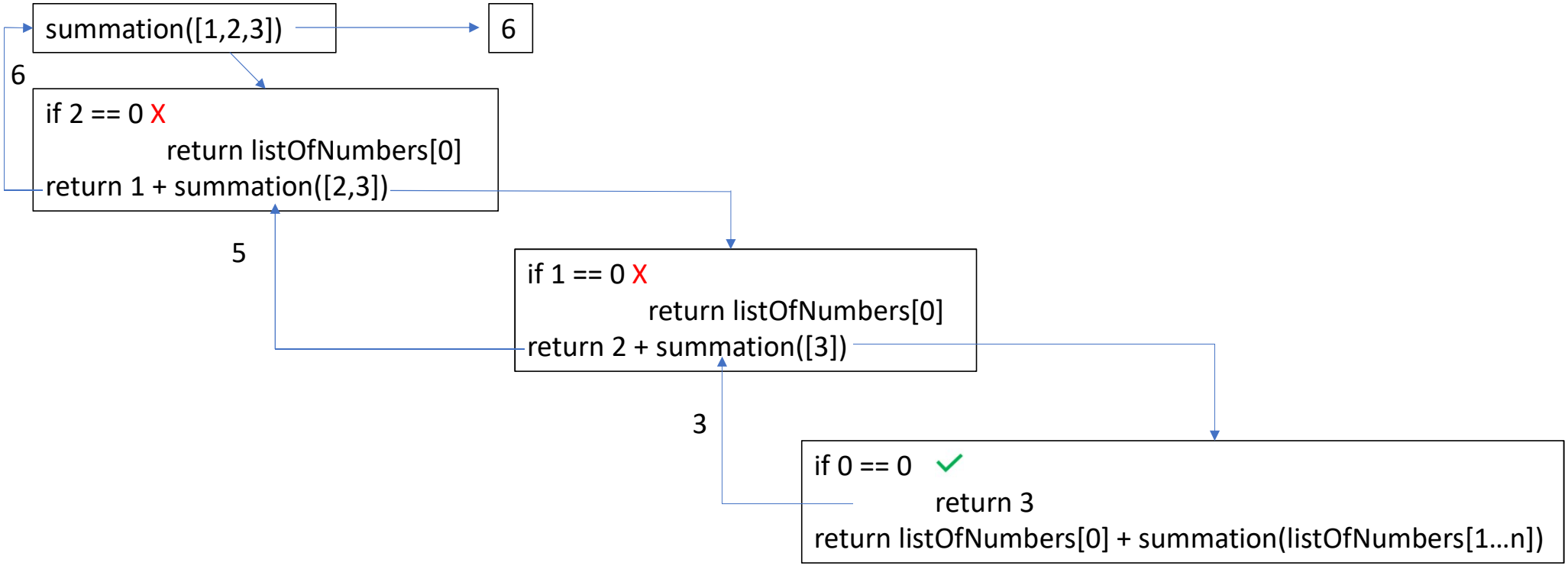
- When a function calls itself one or more times (directly or indirectly)
- Form of repetition
- Typically used to perform same operation on a smaller subset and then build the result based on what is returned from the smaller case
- Typically has at least one base case for stopping
- Based on inductive logic

Iteration vs. Recursion

- Anything that can be done iteratively can be do recursively and vice versa
  - Not always a good idea, some problems naturally lend themselves to one mode of thinking or the other

```
summation(listOfNumbers[0…n])
        if n == 0
                return listOfNumbers[0]
        return listOfNumbers[0] + summation(listOfNumbers[1…n])
```

How it works on a high level



Pros and Cons

- Pros
  - Readable
  - Sometimes easier to conceptualize for problems that have many moving parts
- Cons
  - Efficiency
  - Memory usage
    - Each call to the function makes a new function stack frame (see previous slide)

Example: Factorial

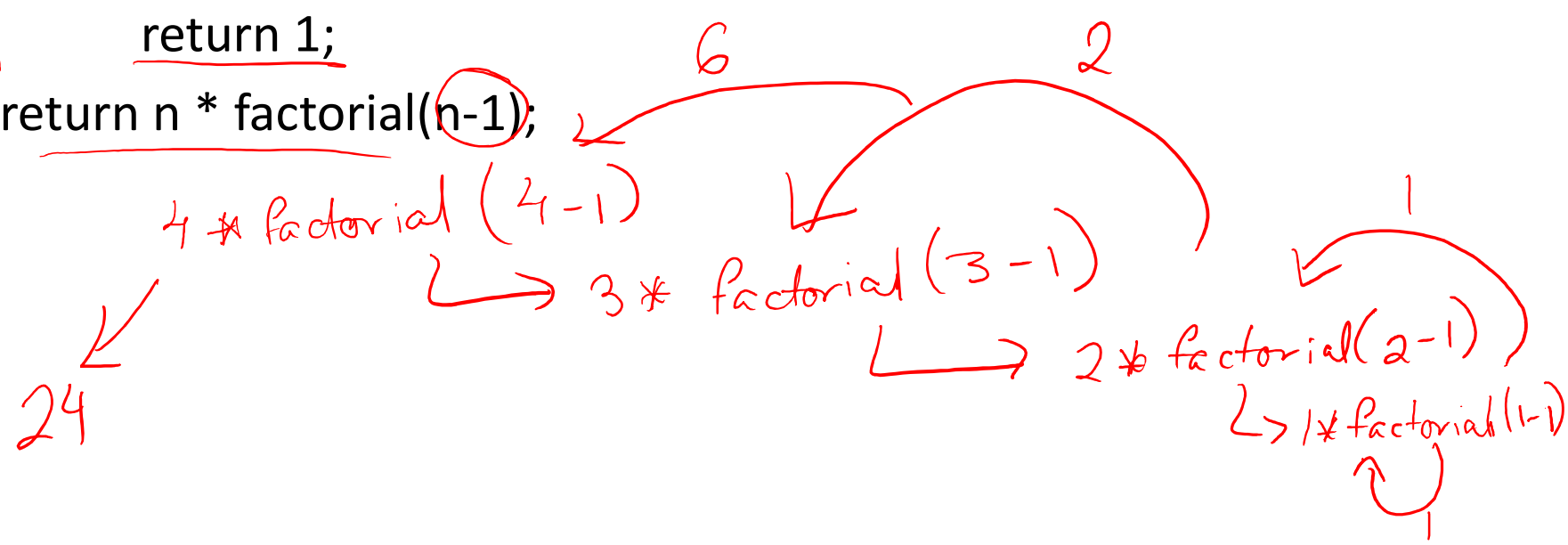
- The product of an integer and all that come before it
- n! = n * (n-1) * (n-2) * … * (n-(n-1)) * 1 for all n > 0
- Base Case: 0! = 1

Iterative Factorial

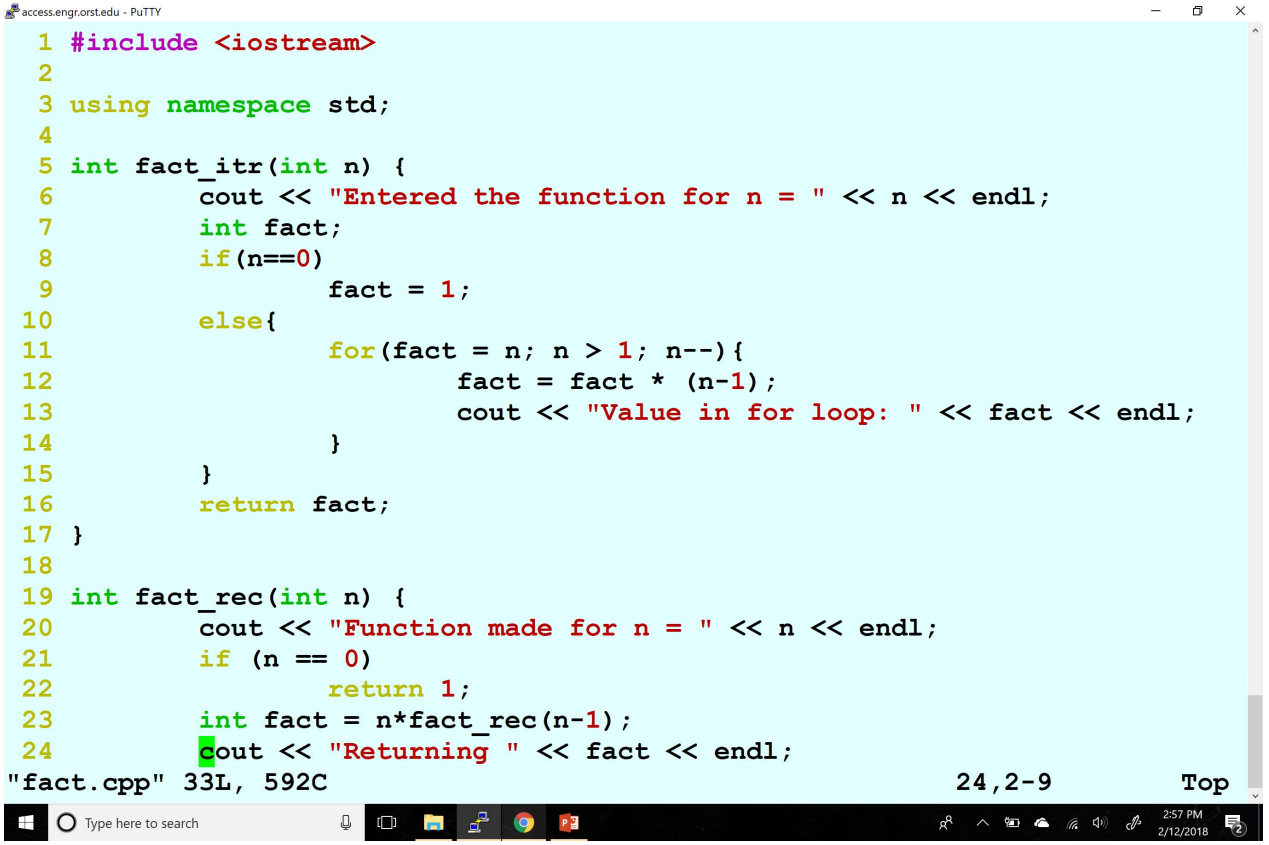
```
int factorial(int n) {
        int fact;
        if (n == 0)
                fact = 1;
        else
                for (fact = n; n > 1; n--)
                        fact = fact * (n-1);
        return fact;
}
```

Recursive Factorial

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



Code Demo


access.engr.orst.edu - PuTTY
1 #include <iostream>
2
3 using namespace std;
4
5 int fact_itr(int n) {
6 cout << "Entered the function for n = " << n << endl;
7 int fact;
8 if(n==0)
9 fact = 1;
10 else{
11 for(fact = n; n > 1; n--){
12 fact = fact * (n-1);
13 cout << "Value in for loop: " << fact << endl;
14 }
15 }
16 return fact;
17 }
18
19 int fact_rec(int n) {
20 cout << "Function made for n = " << n << endl;
21 if (n == 0)
22 return 1;
23 int fact = n*fact_rec(n-1);
24 cout << "Returning " << fact << endl;
"fact.cpp" 33L, 592C
24,2-9
Top
Type here to search
2:57 PM
2/12/2018

access.engr.orst.edu - PuTTY

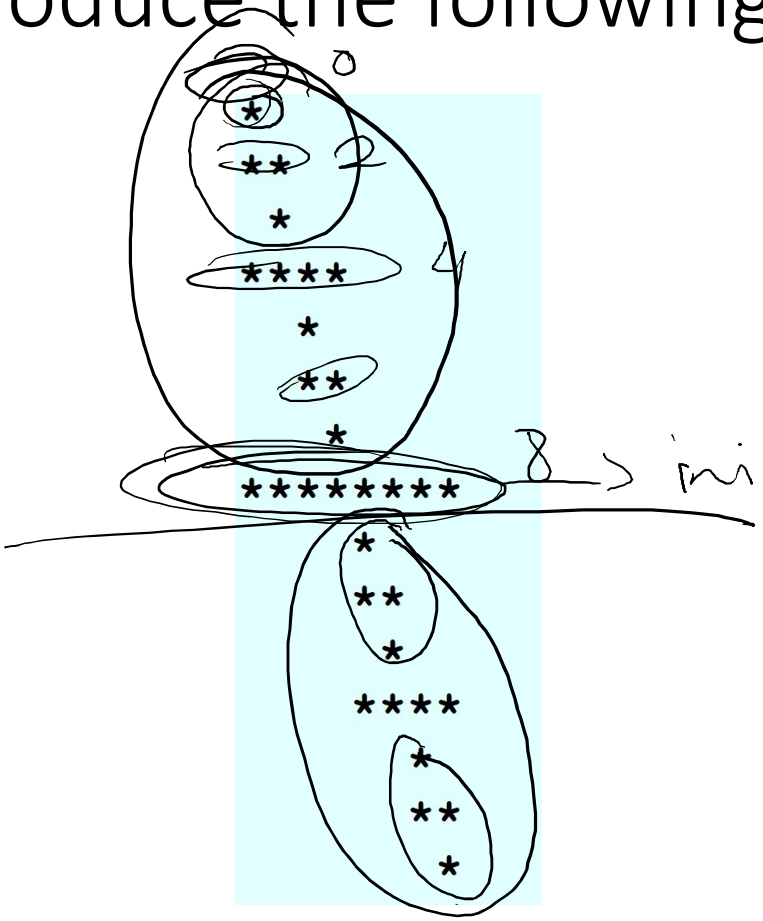
```
10          else{
11                  for(fact = n; n > 1; n--){
12                          fact = fact * (n-1);
13                          cout << "Value in for loop: " << fact << endl;
14                  }
15          }
16          return fact;
17  }
18
19  int fact_rec(int n) {
20          cout << "Function made for n = " << n << endl;
21          if (n == 0)
22                  return 1;
23          int fact = n*fact_rec(n-1);
24          cout << "Returning " << fact << endl;
25          return fact;
26  }
27
28  int main() {
29
30          cout << "Fact iter: " << fact_itr(4) << endl;
31          cout << "Fact rec: " << fact_rec(4) << endl;
32          return 0;
33  }
```

24,2-9 Bot

Type here to search

2:58 PM
2/12/2018

Exercise: What is the recursive solution to produce the following design?




```
if (n == 1
cout << "*" << endl;
func( 8, 0)
      #stars  offset
  ↳ 8/2
    ↳ 4/2
      ↳ 2
```

```
print ( stars, col)
    for # of cols
        cout << " ";
    for # of stars
        cout << "*";
    cout << endl;
```

```
if ( stars == 2)
    print (star, col)
else
```

access.engr.orst.edu - PuTTY

```
 1 #include <iostream>
 2
 3 using namespace std;
 4
 5 void print(int stars, int col) {
 6         for(int i=0; i<col; i++)
 7                 cout << " ";
 8         for(int i=0; i<stars; i++)
 9                 cout << "*";
10         cout << endl;
11 }
12
13 void pattern(int stars, int col) {
14         if (stars == 2){
15                 print((stars/2), col);
16                 cout << "Top if 2" << endl;
17         }
18         else{
19                 cout << "Top else" << endl;
20                 pattern((stars/2), col);
21                 cout << "returned" << endl;
22         }
23         cout << "Mid" << endl;
24         print(stars, col);
```

24,2-9 Top

Type here to search
2:58 PM
2/12/2018

access.engr.orst.edu - PuTTY

```
13 void pattern(int stars, int col) {
14         if (stars == 2){
15                 print((stars/2), col);
16                 cout << "Top if 2" << endl;
17         }
18         else{
19                 cout << "Top else" << endl;
20                 pattern((stars/2), col);
21                 cout << "returned" << endl;
22         }
23         cout << "Mid" << endl;
24         print(stars, col);
25         if (stars == 2){
26                 cout << "Bottom if 2" << endl;
27                 print((stars/2), col+(stars/2));
28         }
29         else{
30                 cout << "Bottom else" << endl;
31                 pattern((stars/2), col+(stars/2));
32                 cout << "returned" << endl;
33         }
34 }
35
36
```

24,2-9 70%

Inbox - ernstsh@oregonstate.edu - Oregon State University Mail - Google Chrome

Type here to search

2:58 PM
2/12/2018

access.engr.orst.edu - PuTTY

```
 22          }
 23          cout << "Mid" << endl;
 24          print(stars, col);
 25          if (stars == 2){
 26                  cout << "Bottom if 2" << endl;
 27                  print((stars/2), col+(stars/2));
 28          }
 29          else{
 30                  cout << "Bottom else" << endl;
 31                  pattern((stars/2), col+(stars/2));
 32                  cout << "returned" << endl;
 33          }
 34  }
 35
 36
 37  int main () {
 38
 39          pattern(8, 0);
 40          return 0;
 41  }
~
~
~
~
                                                        24,2-9          Bot
```

Type here to search

2:58 PM
2/12/2018