Recursion

10-23-2002

Opening Discussion

- What did we talk about last class?
- Do you have any questions about the assignment? Don't let problems hang you up too long.
- What happens in C when a function calls another function? What about when a function calls itself? Can we utilize this to do interesting things?

Min Element of 2D Array

- Return does not mean print.

```
int minElem(int a[][10],int len) {
     int min=a[0][0],j,k;
     for(j=0; j<len; j++) {
          for(k=0; k<10; k++) {
               if(a[j][k]<min) {
                    min=a[j][k];
               }
          }
     }
     return min;
}
```

Encrypt and Decrypt

- Let's look really quick at possible implementations of encrypt and decrypt from assignment #4.
- Note the use of a static variable to keep track of how many characters had been read in the two functions. Outside of that, the logic is fairly simple, but it has to be done in the right order.

Recursive Functions

- We have looked at functions calling other functions, but what about when they call themselves? A function that calls itself is called a recursive function. They are heavily used in the mathematical theory of CS, but they can also be incredibly powerful tools in our code.
- There are some problems that are easy to solve with recursion, but extremely hard to solve without it.

Recursion and the Stack

- The real key to recursion is that one the stack, every call to the function gets its own stack frame and hence its own copy of all the local variables and arguments.
- This gives recursion some "memory" that it can go back to when functions return.
- Languages that don't use a stack for function calls can't be used for writing recursion.

Looping with Recursion

- The simplest use of recursion is simply to implement a loop. When doing this, typically a function takes a value as an argument and calls itself with a larger or smaller value of that argument.
- The recursive call has to be conditional. Otherwise it is like an inifinite loop, but this type causes a stack overflow and a seg fault.

Recursion in Multiple Directions

- Recursion is more powerful (and more useful) when the function contains two or more possible calls of itself. In this case, a loop can't be easily used as a substitute without major logic modifications.
- Drawing out the call path of these types of functions produces what computer scientists call a tree.
- Normally I do Fibonacci numbers for this.

Flood Fill

- If you have used a painting package, you are familiar with the idea of "filling" the area around a specific location with a certain color.
- One way of doing this easily in code is with a recursive function that calls itself up to 4 times if adjacent pixels are blank.
- Let's write a bit of code for simple loops and this.

Minute Essay

- Write a recursive function to calculate Fibonacci numbers.
- Enjoy fall break and I will see you next week. I'll actually be out of town from Thursday afternoon until Sunday and it is very unlikely that I will be able to get to e-mail so you really need to get your questions to me by early tomorrow.