ECE264: Advanced C Programming

Summer 2019

Week 4: Recursion (contd..), File handling, Sorting, Dynamic Data Structures

Recursion – a real life example

"This is an increasingly common occurrence in our political **discourse**."
Washington Post Jun 25, 2019

discourse:
a formal discussion of a subject in speech or writing, as a dissertation, **treatise**, sermon, etc.

treatise:
a formal and systematic **exposition** in writing of the principles of a subject, generally longer andmore detailed than an essay.

exposition:
the act of **expounding**, setting forth, or explaining.

expound:
To set forth or state in detail

```
void LookUpDictionary(string n) {
      array<string> retVal = GetMeaning(n)
      foreach element in retVal:
            if meaning of element is known
                  continue;
            else
                  LookUpDictionary(element);
}
```

Example - Factorial

- n! = n x (n-1) x (n-2) x . . . x 3 x 2 x 1

  (n–1)! = (n-1) x (n-2) x . . . x 3 x 2 x 1

  therefore,

  n! = n x (n-1)!

  is this complete?

- plug 0 to n and the equation breaks.

therefore,

$$n! = \begin{cases} n \times (n-1)! & \text{when } n \geq 1 \\ 1 & \text{when } n=0 \text{ // factorial of negative numbers not defined.} \end{cases}$$

Example - Factorial

$$n! = \begin{cases} n \times (n-1)! & \text{when } n \geq 1 \\ 1 & \text{when } n = 0 \end{cases}$$

// factorial of negative numbers not defined.

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

Example - Factorial

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

Exercise

```
1 int ex1(char* str)
2 {
3     if(*str == ‘\0’)
4           return 0;
5     else
6           return 1 + ex1(str+1);
7 }
```

what does the function ex1 do?

Using gdb to understand recursion

- Demo

```
#include<stdio.h>
int foo(int n)
{
	int retval = n;
	if (n == 0)
		return 1;
	retval = retval * foo(n-1);
	return retval;
}

int main()
{
	int x = foo(5);
	printf(“foo(5)=%d\n”,x);
}
```

Exercise

- What happens in memory when recursion never terminates?

Tail Recursion

```
void printStars(int n) {
      if(n ==1)
            return;
      printf(“*”);
      printStars(n-1);
}
```

- Recursive call is the last statement in the function

Optimizing Tail Recursion

```
void printStars(int n) {
start: if(n ==1)
           return;
       printf(“*”);
       n=n-1;
       goto start;
}
```

- Recursive call replaced by goto statement

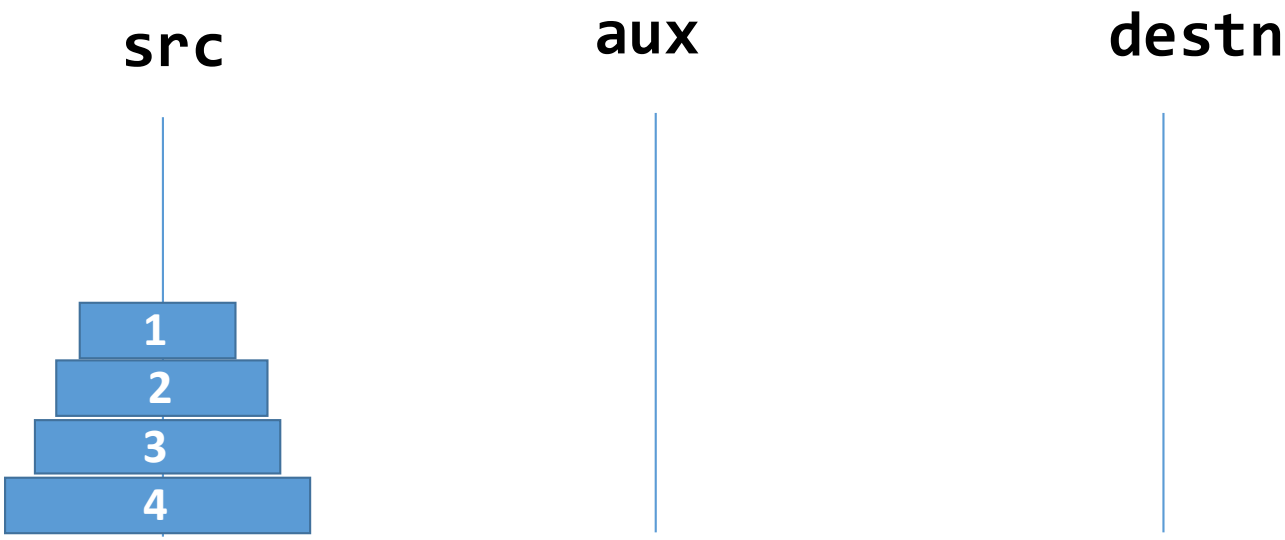
Divide-and-conquer – a common recursive pattern

- A problem can be broken into two or more smaller problems of similar or related type

  Array sum – a toy example

  Quicksort, Mergesort – realistic examples

Tower of Hanoi



1. Move (n-1) disks from `src` to aux (using `destn`)
2. Move disk n from `src` to `destn`
3. Move (n-1) disks from aux to `destn` (using `src`)

Tower of Hanoi – recursive code skeleton

```
void TOH(int n, Rod src, Rod destn, Rod aux) {


      TOH(n-1, src, aux, destn);

      print(“Move disk n from rod <src> to <destn>”);

      TOH(n-1, aux, destn, src);
}
```

Tower of Hanoi – recursive code base case

```
void TOH(int n, Rod src, Rod destn, Rod aux) {


      TOH(n-1, src, aux, destn);

      print("Move disk n from rod <src> to <aux>");

      TOH(n-1, aux, destn, srcdestn);
}
```

- **if n = 0**

  no work to do!

Tower of Hanoi – recursive code base case

```
void TOH(int n, Rod src, Rod destn, Rod aux) {
      if(n == 0)
            return;
      TOH(n-1, src, aux, destn);

      print("Move disk n from rod <src> to <aux>");

      TOH(n-1, aux, destn, srcdestn);
}
```

Tower of Hanoi – analysis

How many steps (`print` statements) do we need to move `n` disks from `src` to `destn` ?

```
void TOH(int n, Rod src, Rod destn, Rod aux) {
      if(n == 0)
            return;
      TOH(n-1, src, aux, destn);

      print("Move disk n from rod <src> to <aux>");

      TOH(n-1, aux, destn, srcdestn);
}
```

Tower of Hanoi – analysis

```
void TOH(int n, Rod src, Rod destn, Rod aux) {
      if(n == 0)
            return;
      TOH(n-1, src, aux, destn);

      print("Move disk n from rod <src> to <aux>");

      TOH(n-1, aux, destn, srcdestn);
}
```