COP 3502 11/30/2024

Exam 1 Avg Scores per Question

1q Q1: 71% ↑ Q5: 90% ✓ expected
1b Q2: 71% ↑ Q6: 51% ✗ ↑
2 Q3: 23% Q7: 60%
4 Q4: 70% ✓ good

Groups!

Study

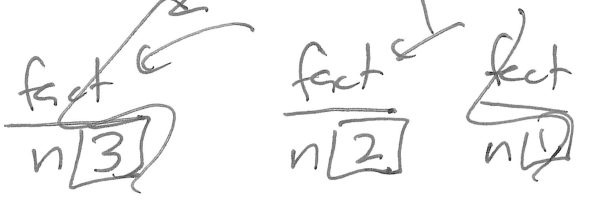
Recursion

a recursive function calls itself sometimes.

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

Base case

$$n! = 1 \times 2 \times 3 \times \dots \times n$$
$$= (1 \times 2 \times \dots \times (n-1)) \times n$$
$$= (n-1)! \times n$$



main

```
int x = fact(3);
```

x 6

~~fact(0)~~
~~fact(1)~~
~~fact(2)~~
~~fact(3)~~
main 6

Gen Strategy:

If input is so easy we can answer easily ⇒ Do so (Base case)

Otherwise:

Break problem down into smaller "pieces" where at least 1 piece is a problem of the exact same nature!

TOWERS

$s \neq e$, $s, e \in \{1, 2, 3\}$

```
towers(int n, int start, int end) {
    if (n > 0) {
        int mid = 6 - start - end;
        towers(n-1, start, mid);
        printf("Move disk %d from
                tower %d to tower %d\n",
                n, start, end);
        towers(n-1, mid, end);
    }
}
```

$T(n) = 2T(n-1) + 1$, $T(1) = 1$

towers recurrence

$T(n) = 2^n - 1$ (result)

Binary Search Recursive

```
                                                    int val
int    search(int* array, int low, int high) {
       if (low > high) return 0;
       int mid = (low+high)/2;
       if( val < array[mid])
           return search(array, low, mid-1);
       else if (val > array[mid])
           return search(array, mid+1, high);
       else
           return 1;
}
```

Fast Modular Exponentiation

$base^{exp} \bmod MOD = ?$

$b^e \% m$ — $O(e)$ is runtime

long long upto 10^{18}

```
                                                long   long   long
long long  slowmodexpo(long b, long e, long m){
       long long res = 1;
       for (long long x = 0; x < e; x++)
           res = (res * b) %m;
       return res;
}
```

Can we do this faster?

$$b^{100} = b^{50} \times b^{50}$$

$$b^{100} \% m = (b^{50} \% m) \times (b^{50} \% m)$$

Same

Key idea: store ans 1st time then reuse

```
ll fastmodexpo(ll b, ll e, ll m) {
    if (e==0) return 1;
    if (e%2==0) {
        ll tmp = fastmodexpo(b, e/2, m);
        return (tmp*tmp)%m;
    }
    return (b * fastmodexpo(b, e-1, m))%m;
}
```

$e \longrightarrow e/2 \longrightarrow e/4$

2 steps 2 steps

$$\frac{e}{2^k} = 1 \Rightarrow k = \log_2 e$$

$$\# \text{ steps} \leq 2 \times \log_2 e$$

$$O(\lg e)$$

$10^{10} \Rightarrow$ no more 80 steps

68 or 70