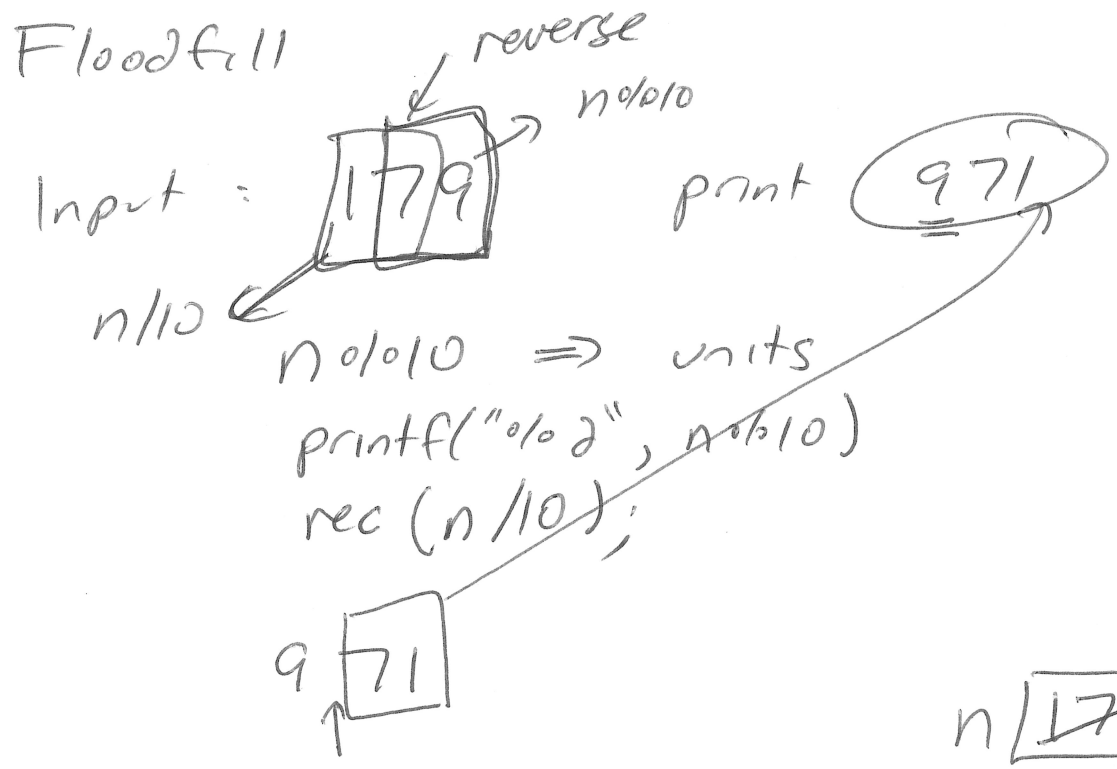
COP3502 9/20/23

Recursion

- ~~Base Conversion~~ Reversing Number (printing)
- Counting occurrences of a letter in string
- Fast Mod Exp
- Floodfill

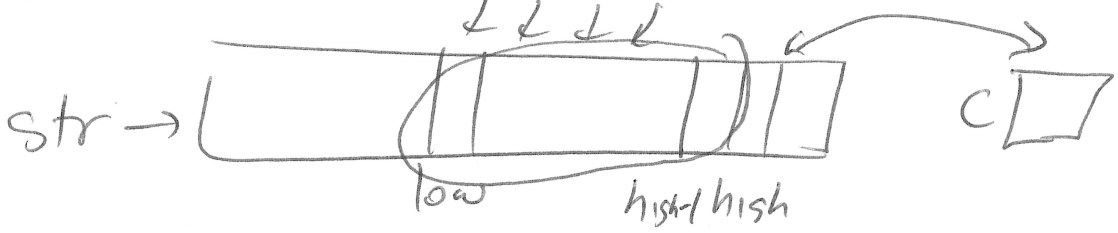


n 179 ~~17~~ 10

rev ~~0~~ 971

```
int rev = 0;
while (n > 0) {
    rev = 10 * rev + n%10;
    n /= 10;
}
```

Count # times char c appears in string str from index low to index high




```
return count(str, low, high-1, c) + (str[high]==c);
```

Fast Mod Expo

b^e % mod

Run time O(exp)

Slow Mod Expo

```
res = 1
for (i=0; i<exp; i++)
    res = (res * base) % mod;
return res;
```

$b^{1000000} = b^{500000} * b^{500000}$

Divide conquer

```
if (exp%2 == 0) {
    tmp = fastmodexpo(base, exp/2, mod);
    return (tmp * tmp) % mod;
}
```

1000000 → 500,000 → 250,000 → 125,000 → 62500
→ 31250 → 15625 → 15624 → 7812

$b^e = b^{e-1} \times b$

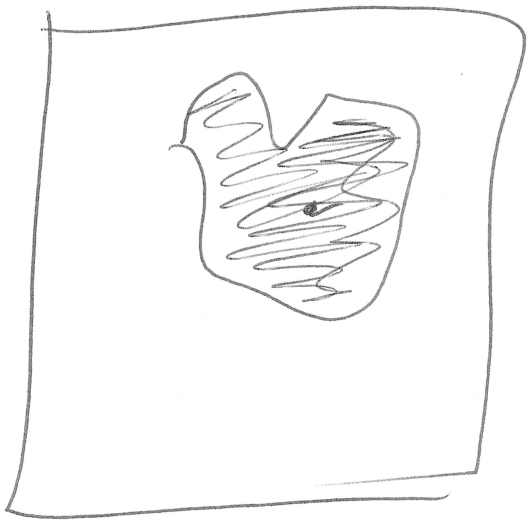
2 steps divide exp by 2

$\frac{exp}{2^k} = 1 \qquad exp = 2^k$

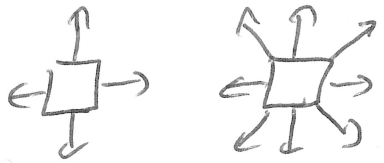
$k = \log_2 exp$

rec calls $\leq 2 \cdot \log_2 exp$

Floodfill



Paint Bucket



```
floodfill ( int x, int y, int** grid used ) {
    if (!inbounds(x,y)) return;
    if (used[x][y]) return;
    if (isboundary(x,y)) return;
    grid[x][y] = fillcolor;
    used[x][y] = 1;
    for each neighbor of (x,y)
        floodfill(x', y', grid, used);
}
```

```
const int DX[4] = {-1, 0, 0, 1};
const int DY[4] = {0, -1, 1, 0};
                  NUMDIR

for (int i = 0; i < 4; i++)
    int nx = x + DX[i];
    int ny = y + DY[i];
    floodfill(nx, ny, ...);
}
```

X 3

Y 5

(2,5)

(3,4)

(3,6)

(4,5)