

6/13/25

1) Use of & in functions

2) If time, Counting Stars (Ver 1)

```
int func(vector<int> nums) {
```

```
    nums[0]++;  $\rightarrow$  // print  
    return nums[0];
```

}

main

v $\rightarrow$ [3|4|5]

f(v) nums $\rightarrow$ [0|1|2|3|5|5]

✓ print(min)

func(min)

print(min) ✓

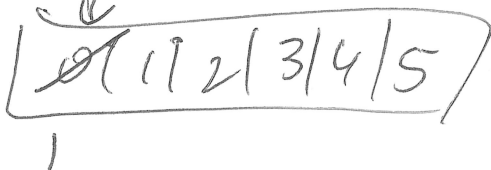
~~func~~
~~nums $\rightarrow$ [3|4|5]~~
~~4~~

~~print~~
~~nums [0|1|2|3|4|5]~~

~~func~~
~~nums $\rightarrow$ [0|1|2|3|4|5]~~
~~1 1 2 3 4 5~~

~~print~~
~~nums $\rightarrow$ [1|1|2|3|4|5]~~
~~1 1 2 3 4~~

`func(vector<int> & nums)` → pass by reference

`func(nums)`
~~nums~~ → 
1

func
nums
nums[0]++;

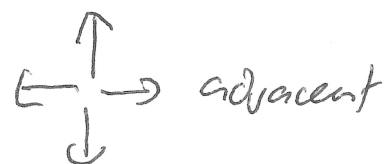
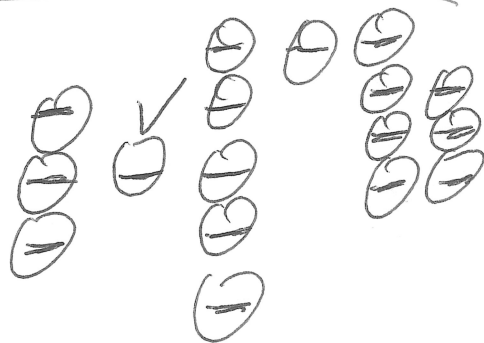
#1) Pass by Value of large objects is very inefficient

#2) We can improve our efficiency by using pass by reference (&)

#3) Pass by value is SAFE because no change can be made to the ~~calling~~ variable from the calling function.

→ When you do pass by reference, be careful of what you change!

Counting Stars



$$DX = \{-1, 0, 0, 1\},$$

$$DY = \{0, -1, 1, 0\},$$

floodfill(int x, int y) {

grid[x][y] = marked;

for (i=0; i<4; i++) {

int newx = x + DX[i];

int newy = y + DY[i];

if (!inbounds(newx, newy))

continue;

if (grid[newx][newy] != '-')

continue;

floodfill(newx, newy);

}