

COP 3502 2/16/22

① Sorted List Matching

* Binary Search
Max Ones

② QUIZ FRI

- early don't sit in 1st 2 rows

Recursion

Linked Lists

Stacks + Queues

SLM

Sorted List Matching

~~Adams~~
~~Baker~~
→ Carter
Davis
Jones → n people

~~Carter~~ Carson
← Carter
Crossin
Davis
Douglas
Ember → m people

Class 1

Class 2

Slow Soln

for each person in class1:

for each person in class2:

if they match
output.

n
m each iteration
→ n * m
times

Doesn't use fact lists are sorted

Better Soln

for each person in class 1: $\rightarrow n$ times
if binarysearch(person, class2) $\rightarrow \log m$ times
Output

In a binary search, we divide the search list by 2 each comparison.

$\frac{m}{2^k} = 1$, $k = \#$ comparisons until algorithm ends

$$2^k = m \Rightarrow k = \log_2 m$$

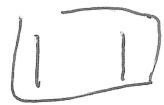
Best Soln (usually)

Start 2 pointers at front of both lists. Advance the one that comes earlier. When there's a match, output it and advance both

$$\# \text{ skip} = n + m$$

Max Ones

array



⇒ 1 0 0 0 0 0 0

1 1 1 ⇒ 1 ⇒ 1 ⇒ 1 ⇒ 0

1 0 0 0 0 0

1 1 1 1 1 0

1 1 1 0 0 0

ans = 0

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

int j = 0;

while (j < cols && array[i][j] == 1)

j++;

if (j > ans) ans = j;

}

WORST CASE: ROWS * COLS

BETTER SOLN

Don't start at beginning of each row!
Skip in the column you left off!

RUNTIME = ROWS + COLS

max # of
1s on a
single row

First Attempt

Target Sum

Input: Sorted Array and integer target

2, 6, 7, 8, 13, 40, 47, 48, 52, 99

target = 48

Slow

```
for (i = 0; i < n; i++)  
    for (j = i + 1; j < n; j++)  
        if (array[i] + array[j] == target)  
            return 1;  
return 0;
```

Binary Search

```
for (i = 0; i < n; i++)  
    if (binarysearch(array, n, target - array[i])  
        return 1;  
return 0;
```

NOTE → need to check that it's not the same value!

Fastest

```
skit low = 0, high = n - 1;  
if (array[low] + array[high] > target  
    high--;  
else if (array[low] + array[high] < target  
    low++;  
else  
    return 1;
```