How is this class different than COP 3223?

---

① We will CARE about the efficiency of a solution! (how fast)

② For grading consistency, we will use Eustis for compiling + running programs.

③ Learn some math to analyze + predict how fast a program will run.

Sorted List Matching Problem

Given 2 separate sorted lists, find all values ~~in~~ that appear in both lists.

List 1: 6, 8, 9, 15, 22, 27, 33, 81, 89 n

List 2: 2, 3, 6, 12, 14, 27, 29 m

```
for (int i = 0 ; i < n ; i++ ) {
    int found = 0;
    for (int j = 0 ; j < m ; j++)
        if (list1[i] == list2[j])
            found = 1;
    if (found) printf("%d\n", list1[i]);
}
```

repeats n times

repeats m times

Runtime = O(mn)

List 1: 6, 6, 9, 15, 22, 27, 33, 81, 85 n

List 2: 2, 3, 6, 12, 14, 27, 29 m

Middle Solution: ~~B~~

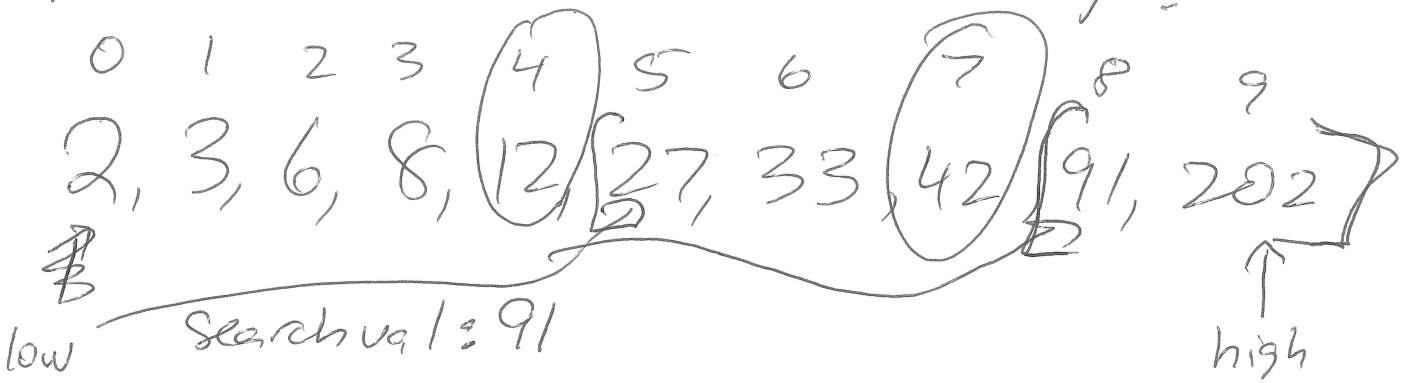
For each number in list 1, search to see if it's in list 2, but use the fact that list 2 is sorted to speed up the search.

Binary Search in a sorted array

What is a binary search? Sec 2

Input: Sorted Array, and a search value

Output: Is the value in the array?



mid = (low+high)/2; (4)

Since 91 > 12, low = 5

mid = (5+9)/2 = 7

Since 91 > 42, low = 8

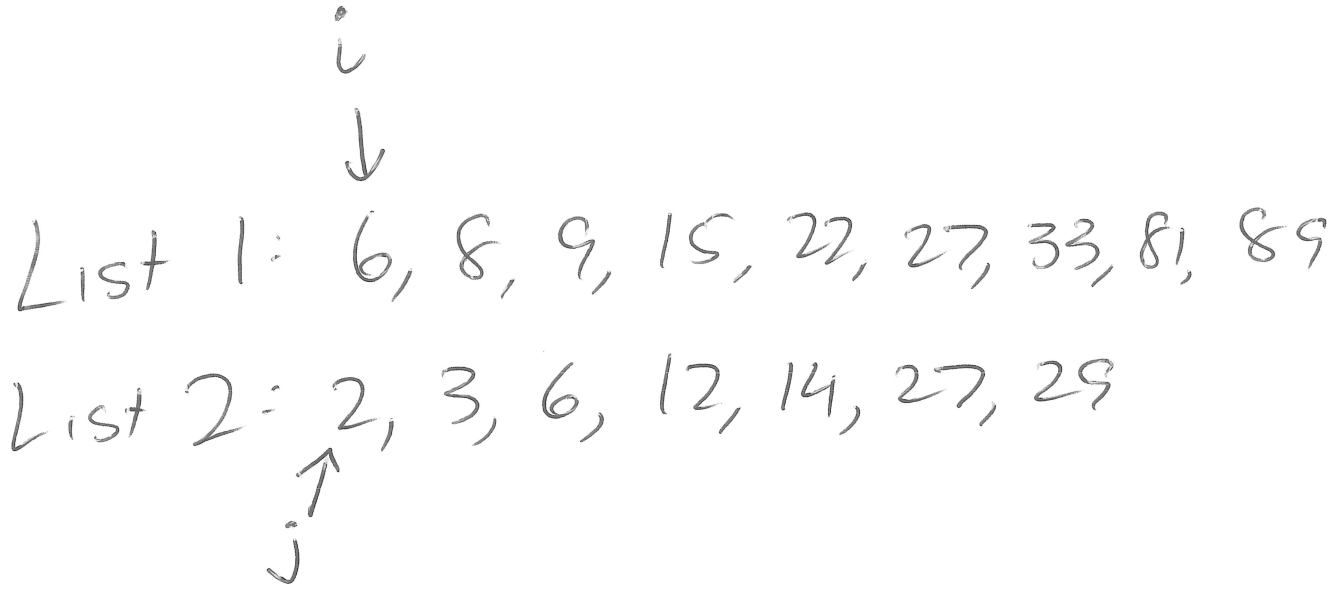
mid = (8+9)/2 = 8 FOUND IT!

for each item in list1

binary search if it's in list2

Run time binary search on n elements is $O(\lg n)$ This algorithm takes $O(n \lg m)$

8/23/21 (S)
Sec1




```
int i=0, j=0;
while (i<n && j<m) {
    if (list1[i] < list2[j]) i++;
    else if (list2[j] < list1[i]) j++;
    else {
        printf("%d\n", list1[i]);
        i++;
        j++;
    }
}
```

Runtime $= O(n+m)$

n	m	nm	n+m
100	100	10,000	200
10^6	10^5	10^{11}	$1{,}100{,}000 = 1.1 \times 10^6$