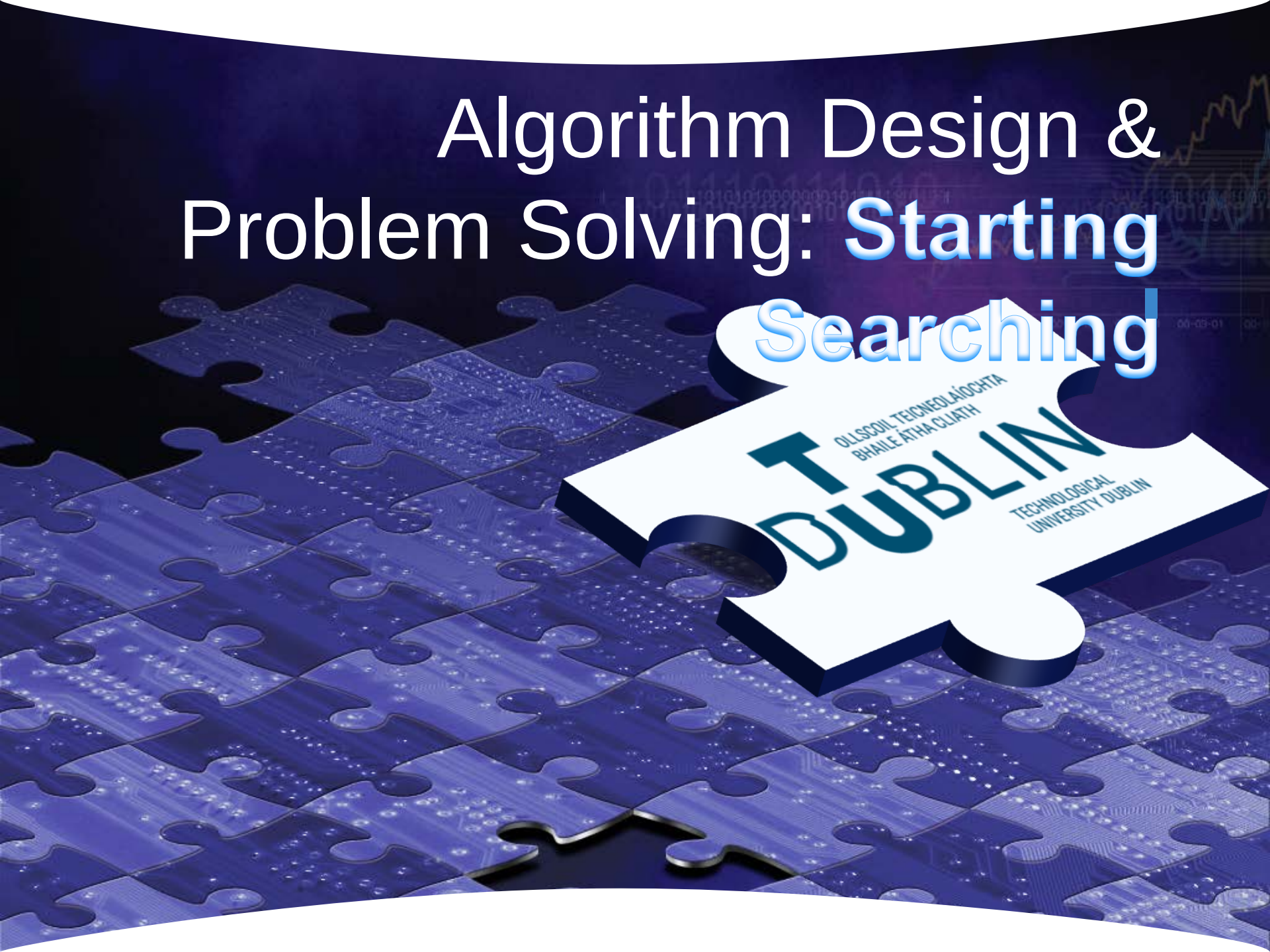
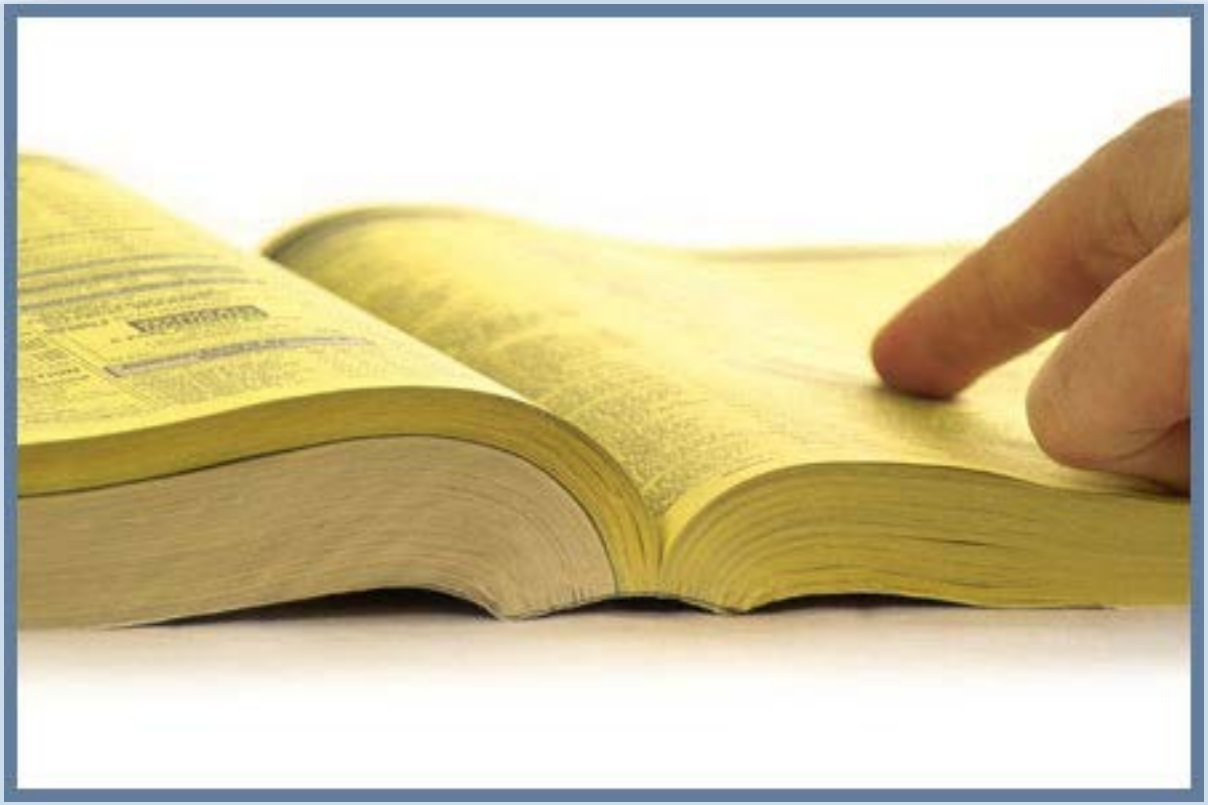
Algorithm Design & Problem Solving: Starting Searching



Contents

How did you find a name in a phonebook?



How did you find this card?

Linear Search

I need to find the student whose name is …..

Linear Search

- Most basic of search algorithms
- Moves **sequentially** through the array looking for a matching value

Linear Search

LINEAR SEARCH		
Best	Average	Worst
O(1)	O(n)	O(n)

Array

Brute-force

Linear Search

- When is it best used?
  - Small problem set
  - Unsorted problem set

Binary Search

1. Everybody stand up
2. Pair off with somebody standing; stay standing if taller, if shorter sit down.
3. If still standing, go to step 2

Divide & Conquer

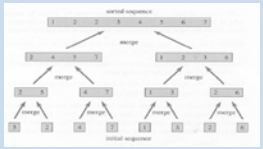
Binary Search

- Relies on a **divide and conquer** strategy to find a value within an already-sorted collection

Binary Search

BINARY SEARCH		
Best	Average	Worst
O(1)	O(log n)	O(log n)

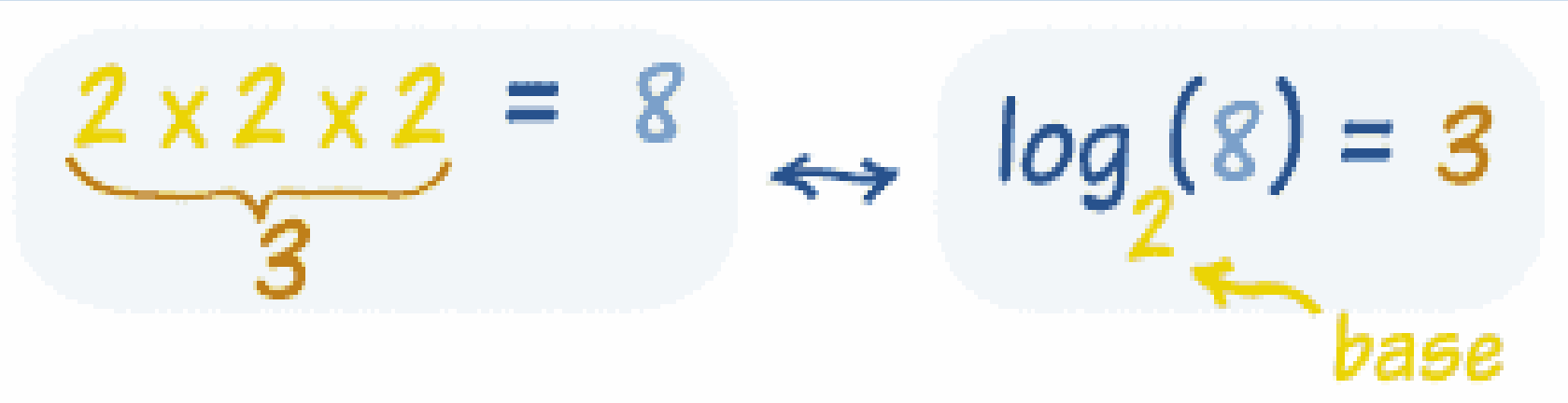
Array



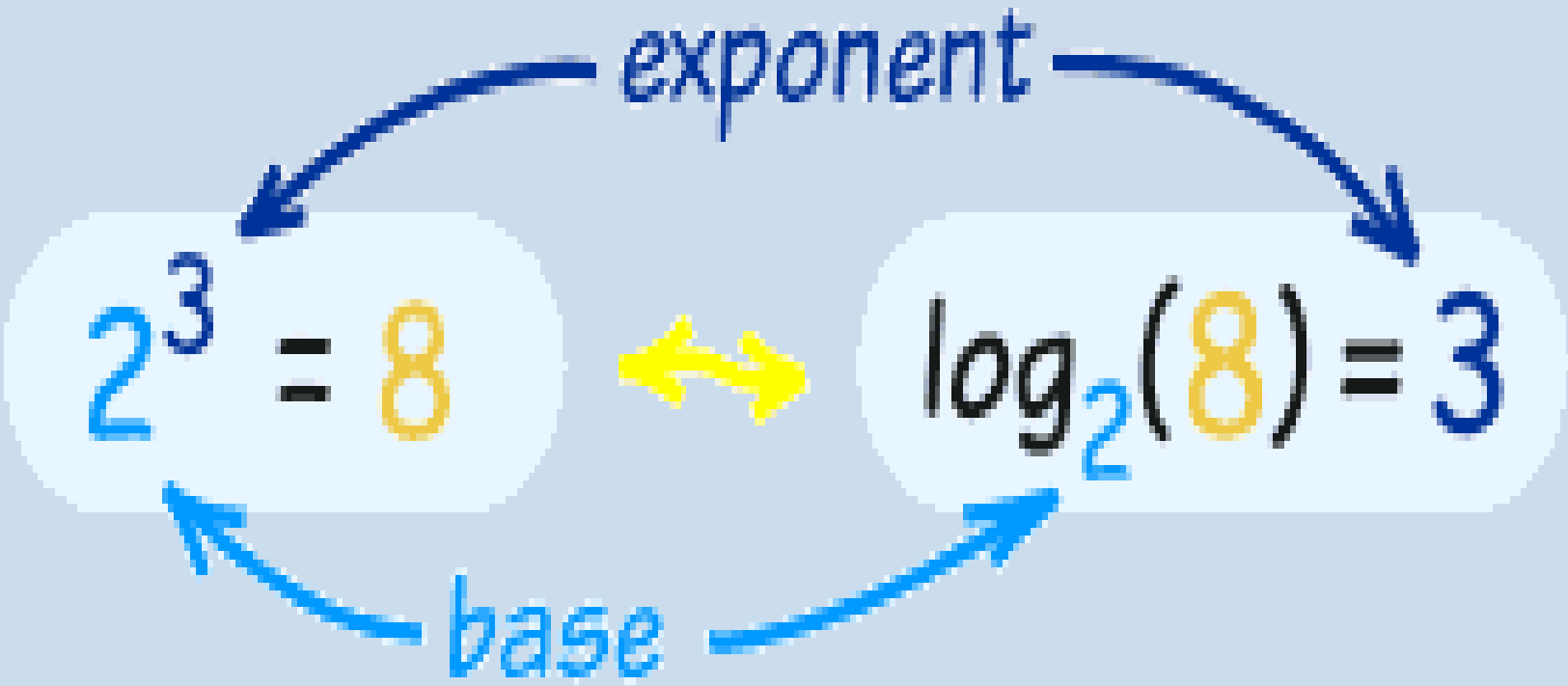
Divide & Conquer

Do you remember logarithms?

- In its simplest form:
  - How many of one number do we multiply to get another number?



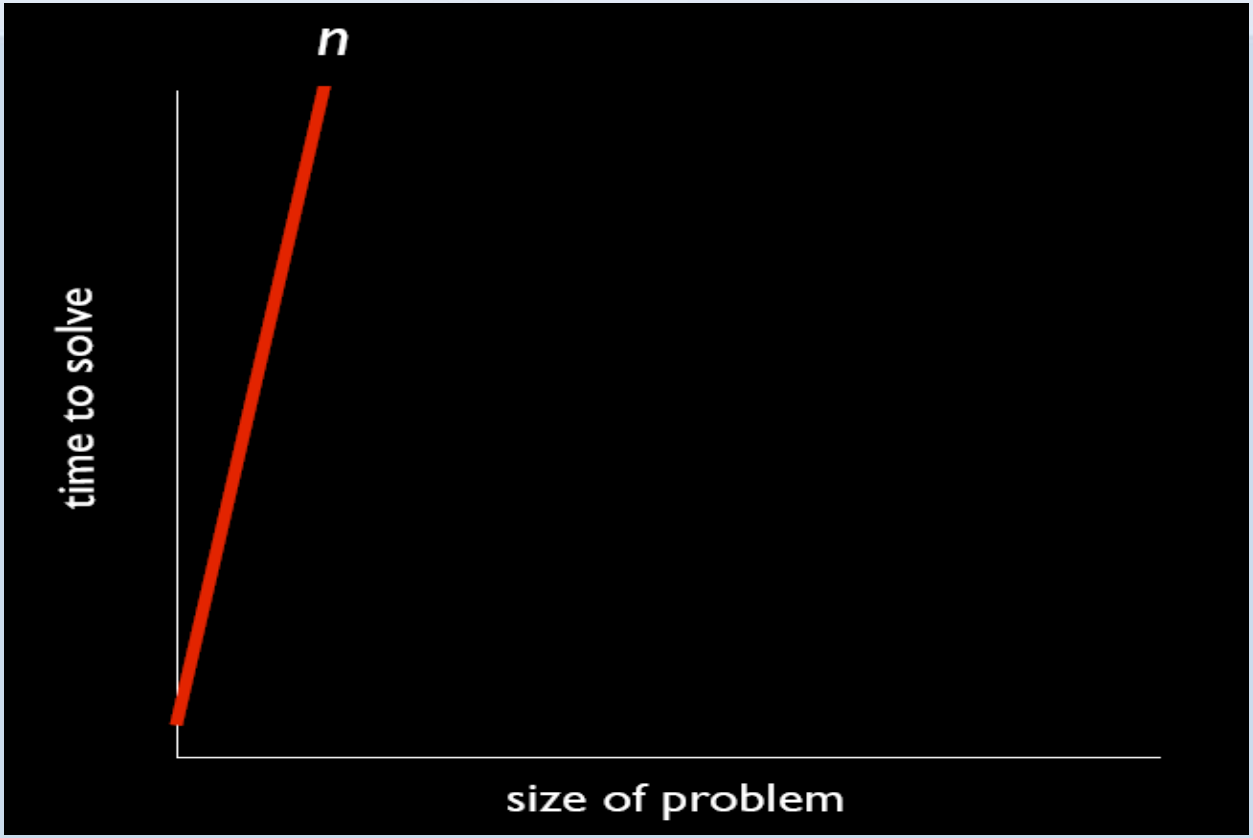
Do you remember logarithms?



Binary Search

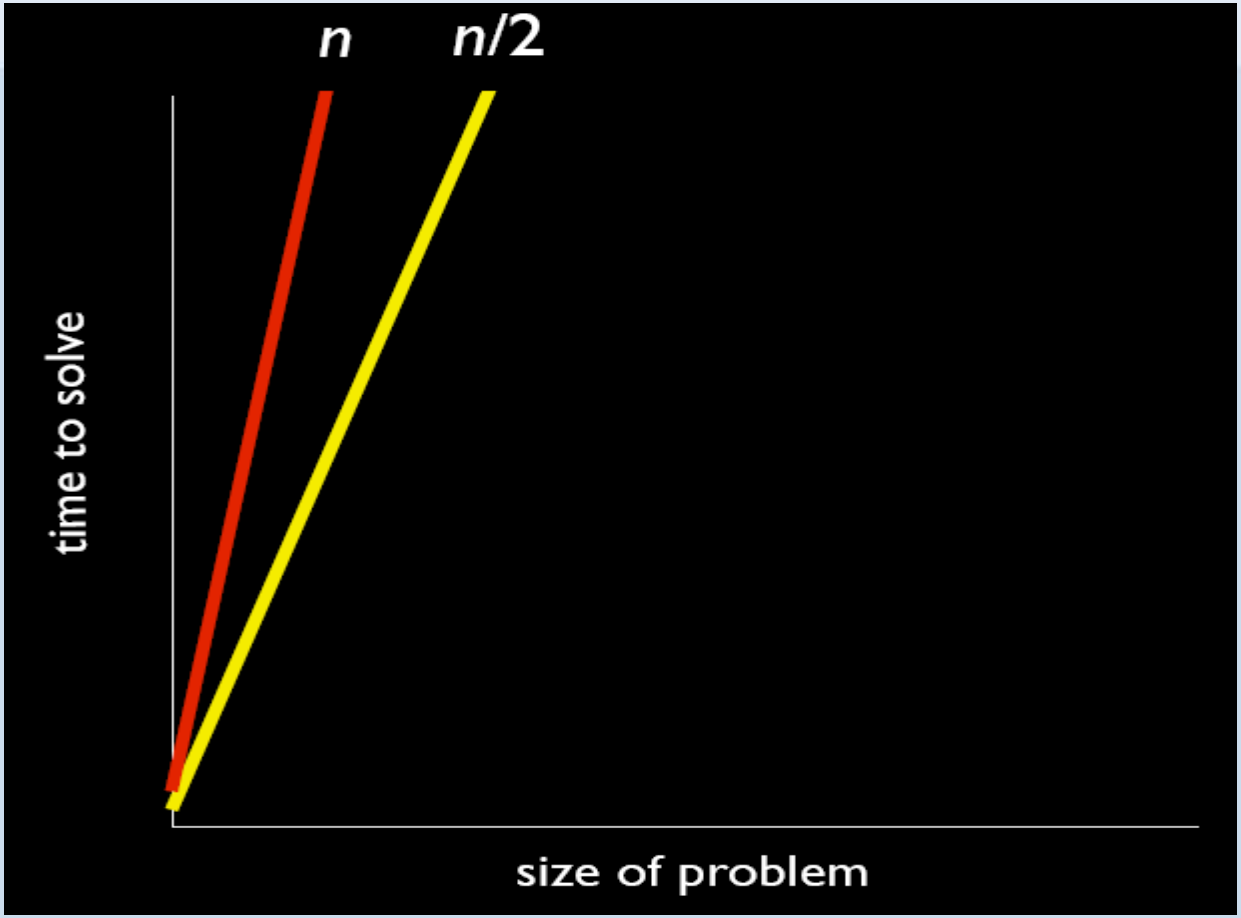
- When is it best used?
  - On a large problem set
  - Sorted problem set

O(n)



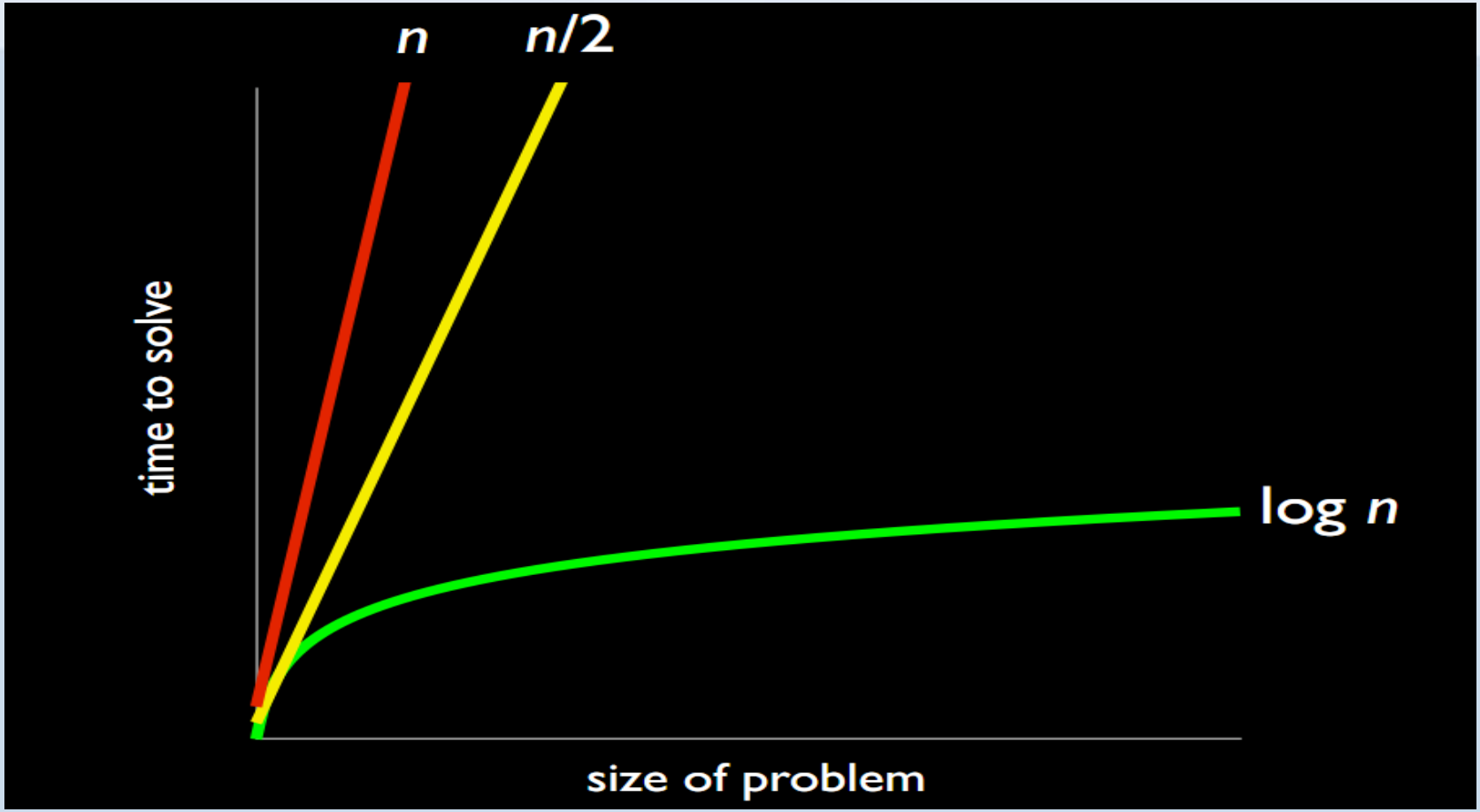
If I count everybody in this class individually - Linear

O(n/2)



If I count the class in pairs

O(Log n)



If I count the class using the Week 1 algorithm - Binary

Search Algorithms

- Write the algorithm to perform a **linear search** on an array.

Linear Search Algorithm

Input: An array **A** storing **N** items**,** element to be searched for - **searchKey**

Output: i, position of the searched for element in the array

Linear Search Algorithm

```
for i = 0, i < n do
    if A[i] = searchKey
        return i
return -1
```

Search Algorithms

- Write the algorithm to perform a **binary search** on an array.

Binary Search Algorithm

1. Determine middle element of subarray
2. If searchKey is the middle element return its index
3. Else if searchKey is in the left subarray
   - Highest index is the index of the middle element – 1
4. Else if searchKey is in the right subarray
   - Lowest index is the index of the middle element + 1

Binary Search Algorithm

Input: An array **A,** element to be searched for – **searchKey,** lowest index of array – **low,** highest index of array – **high.**

Output: i, position of the searched for element in the array

Binary Search Algorithm

Middle = (low + high)/2

If searchKey = A[middle]

return middle

Else if searchKey < A[middle]

high = middle - 1

Else if searchKey > A[middle]

low = middle + 1

What is wrong with this algorithm?

Binary Search Algorithm

```
While low <= high
    Middle = (low + high)/2
    If searchKey = A[middle]
        return middle
    Else if searchKey < A[middle]
        high = middle - 1
    Else if searchKey > A[middle]
        low = middle + 1
```

Binary Search Algorithm

Alter your algorithm to find this card in a pack of playing cards.

Order of suits are:

- Hearts
- Diamonds
- Clubs
- Spades

Thank You !


OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN