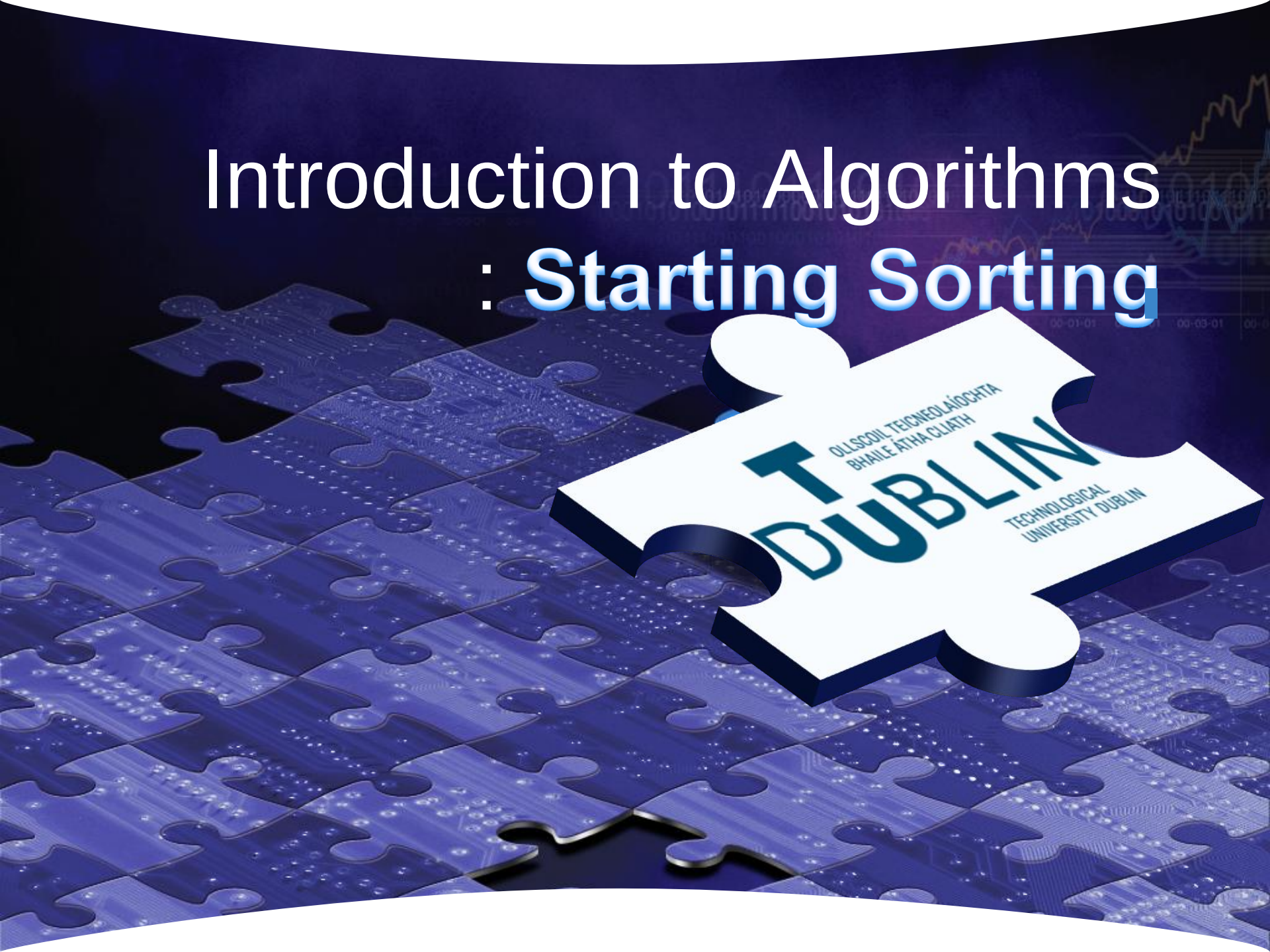
Introduction to Algorithms
: Starting Sorting



Contents

Quiz ToH Q

What does the second recursive call in the TOWER OF HANOI algorithm do?

a) Requesting to move a particular disk from the same tower
b) Requesting to move a particular disk from a different tower
c) None of the above

What does the first recursive call do in the TOWER OF HANOI algorithm?

a) Requesting to move a particular disk from the same tower
b) Requesting to move a particular disk from a different tower.
c) Requesting to move a particular disk from a different tower.

```
1  moveTower (disks, source, dest, spare)
2  If disk = 0
3       Move disk from source to dest
4  Else
5       moveTower (disk-1, source, spare, dest
6       move disk from source to dest
7       moveTower (disk-1, spare, dest, source)
```

Revision

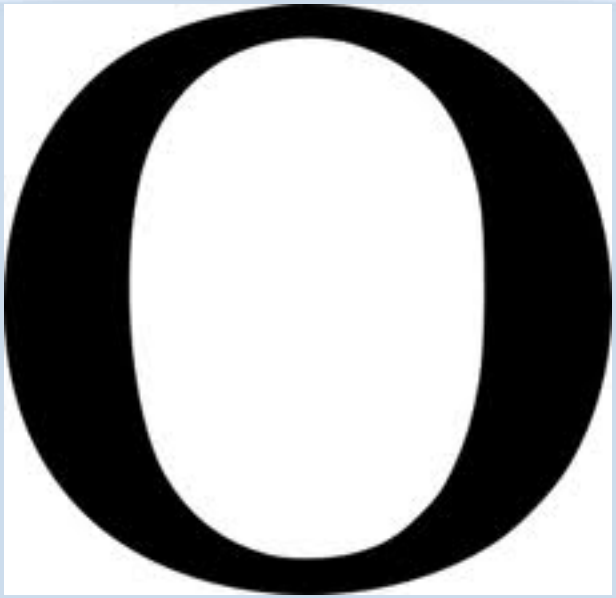
- **An array is a finite, ordered set of homogeneous elements.**

Finite: there is a specific number of elements in the array

Ordered: elements of the array are arranged so that there is a zeroth, first, second ..

Homogeneous: all elements in the array must be of the same type

Big O



- Describes the performance or complexity of an algorithm.
- Describes the **worst-case** scenario - execution time required or the space used by an algorithm.

Big O

$O(1)$

$O(n)$

$O(n^2)$

....

http://rob-bell.net/2009/06/a-beginners-guide-to-big-o-notation/

Introduction to Logarithms

How many of *one number* do we multiply to get *another number*?

How many 2s do we multiply to get 8?

Answer: **2 × 2 × 2 = 8**, so we needed to multiply 3 of the **2**s to get **8**

So the logarithm is 3

$$\underbrace{2 \times 2 \times 2}_{3} = 8 \quad \leftrightarrow \quad \log_2(8) = 3$$

base

http://www.mathsisfun.com/algebra/logarithms.html

Answer the following ….

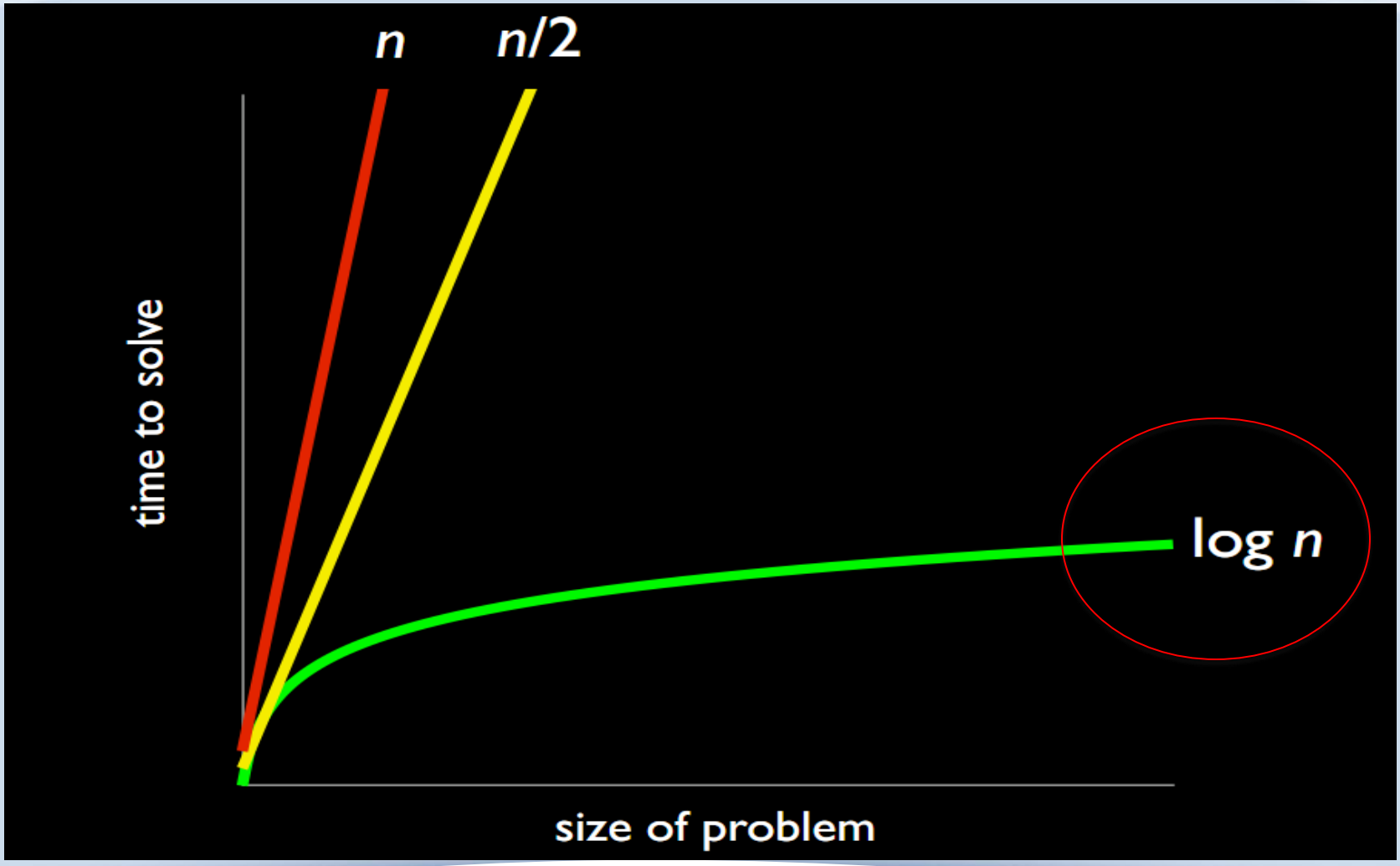
$\log_2(64) =$

$\log_5(625) =$

$\log_{10}(100) =$

$\log_3(81) =$

Where do I need log?



Selection Sort

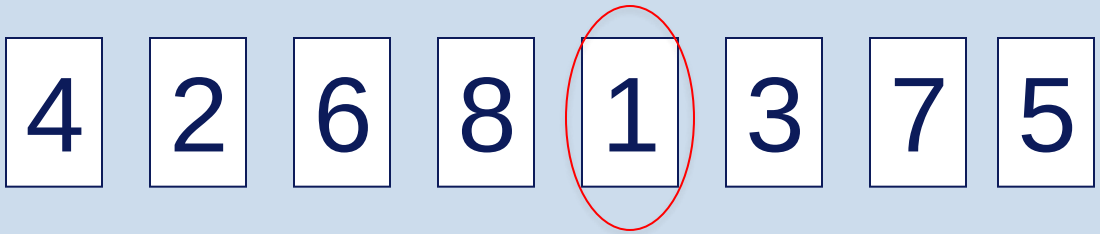
- How can this be sorted using Selection Sort?

4	2	6	8	1	3	7	5

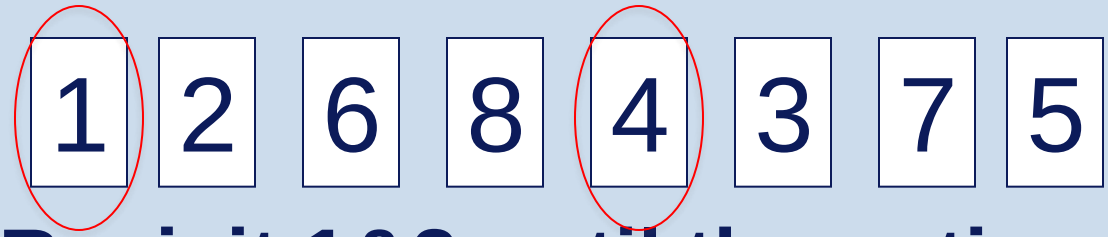
Selection Sort

- What are the steps?

1. Find the smallest unsorted number



1. Swap it into lowest unsorted position



2. Revisit 1&2 until the entire array is sorted

Selection Sort

- How efficient is it?

$$O(n^2)$$

- What does that mean?

Selection Sort

SELECTION SORT		
Best	**Average**	**Worst**
$\Omega(n^2)$	$\Theta(n^2)$	$O(n^2)$

Array

Selection Sort: The algorithm

- If the performance of selection sort is **$O(n^2)$,** this insinuates that there are two loops:

Loop 1
Loop 2
End Loop 2
End Loop 1

n * n

Selection Sort: The algorithm

- What does Loop 2 do?
- What does Loop 1 do?

Loop 1
Loop 2
End Loop 2
End Loop 1

Selection Sort: The algorithm

- What does Loop 2 do?
  - Finds the lowest number
- The new lowest number is then swapped

Selection Sort: The algorithm

- What does Loop 1 do?
  - Loop 2 to find the lowest number
  - Swap numbers

Repeats this n times

Selection Sort: The algorithm

- Can we refine this description?

Repeat Loop 1 N-1 times

Repeat Loop 2 until

Find smallest number

Swap with current number

End Loop 1

Now write the pseudo code for this algorithm

Selection Sort: The algorithm

Input: An array **A** storing **N** items

Output: A sorted in ascending order

Algorithm Selection_Sort (A, N):

```
for i = 0 to i < N-1 do
    min = i
    for j = i+1 to j < N do
        if A[j] < A[min] then
            min = j
    temp = A[min]
    A[min] = A[i]
    A[i] = temp
End for
```

Find the smallest

Make the swap

Could I make this more efficient?

Bubble Sort

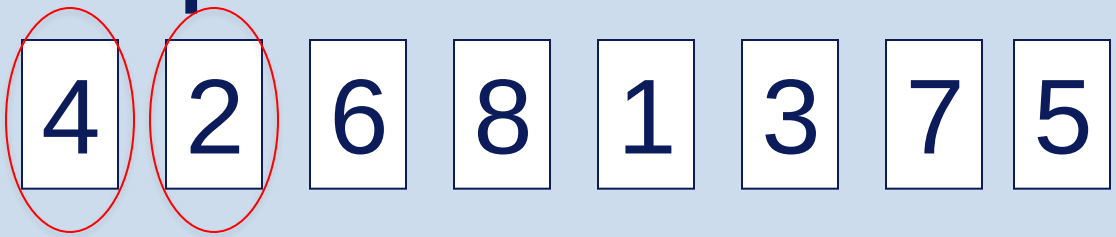
- How can this be sorted using bubble sort?

4	2	6	8	1	3	7	5

Bubble Sort

- What are the steps?

1. Compare first 2 numbers



1. Swap, if necessary, into chronological order

2. Compare 2nd number of Step 2 and next number. Revisit 2 & 3 until completely sorted.

Bubble Sort

1. **Hungarian Folk Dance**

Bubble Sort

- How efficient is it?

$$O(n^2)$$

- Why?

Bubble Sort

BUBBLE SORT		
Best	Average	Worst
$\Omega(n)$	$\Theta(n^2)$	$O(n^2)$

Array

Bubble Sort: The algorithm

- If the performance of bubble sort is **$O(n^2)$,** this insinuates that there are two loops (like selection sort):

Loop 1

Loop 2

End Loop 2

End Loop 1

Bubble Sort: The algorithm

- What does Loop 2 do?
- What does Loop 1 do?

Loop 1
Loop 2
End Loop 2
End Loop 1

Bubble Sort: The algorithm

- What does Loop 2 do?
  - Swaps numbers if necessary

Bubble Sort: The algorithm

- What does Loop 1 do?
  - Loop 2 to swap numbers if necessary

Repeats this n times

Bubble Sort: The algorithm

- Can we refine this description?

Repeat Loop 1 N -1 times

Repeat Loop 2 N -1 times

Swap number if necessary

End Loop 1

Now write the pseudo code for this algorithm

Bubble Sort: The algorithm

Input: An array **A** storing **N** items

Output: A sorted in ascending order

Algorithm Bubble_Sort (A, N):

```
for i = 0 to i < N-1 do
   for j = 0 to j < N-1 do
        if A[j] > A[j+1]
             temp = A[j]
             A[j] = A[j+1]
             A[j+1] = temp
        End if
   End for
End for
```

Make the swap

Recap

- Selection sort algorithm:
  - What are the 3 steps?
  - What is the worse case scenario?
  - How many loops are in this algorithm?
  - What are they for?

Recap

- Bubble sort algorithm:
  - What are the 3 steps?
  - What is the worse case scenario?
  - How many loops are in this algorithm?
  - What are they for?

Thank You !


OLLSCOIL TEICNEOLAÍOCHTA
BHAILE ÁTHA CLIATH
DUBLIN
TECHNOLOGICAL
UNIVERSITY DUBLIN