This is CS50

Week 3

Yuliia Zhukovets

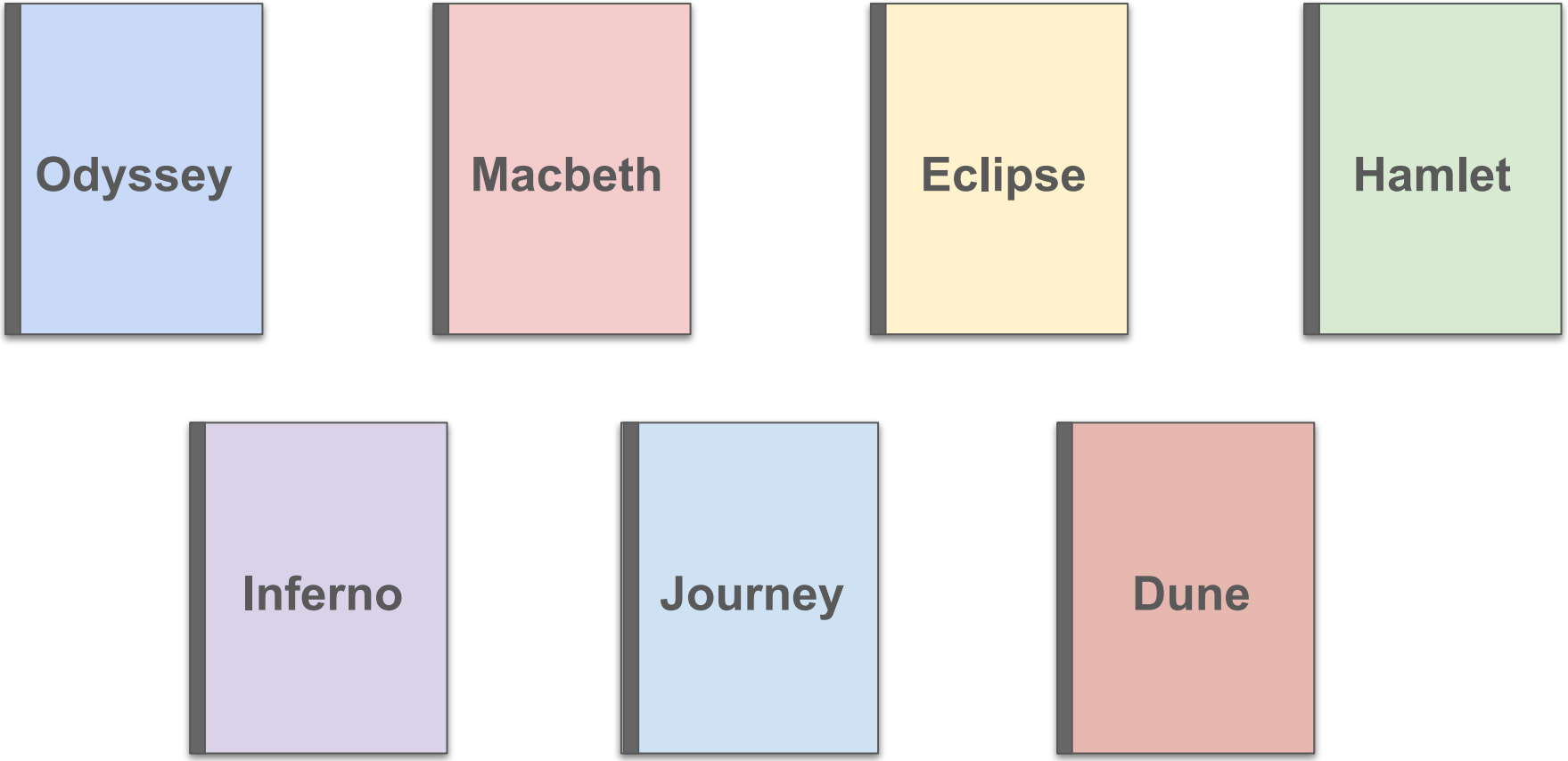
Preceptor

yuliia@cs50.harvard.edu

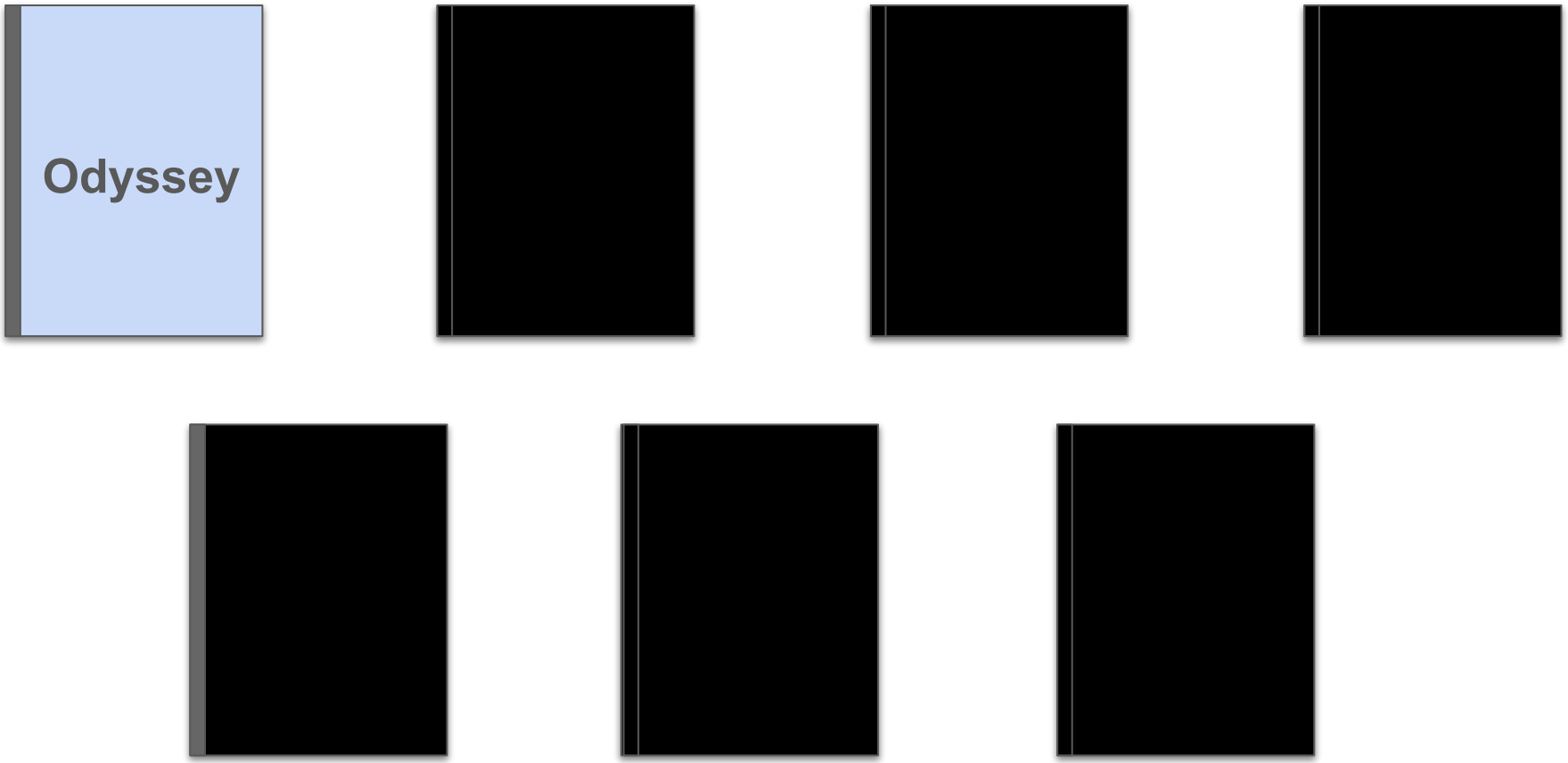
Agenda

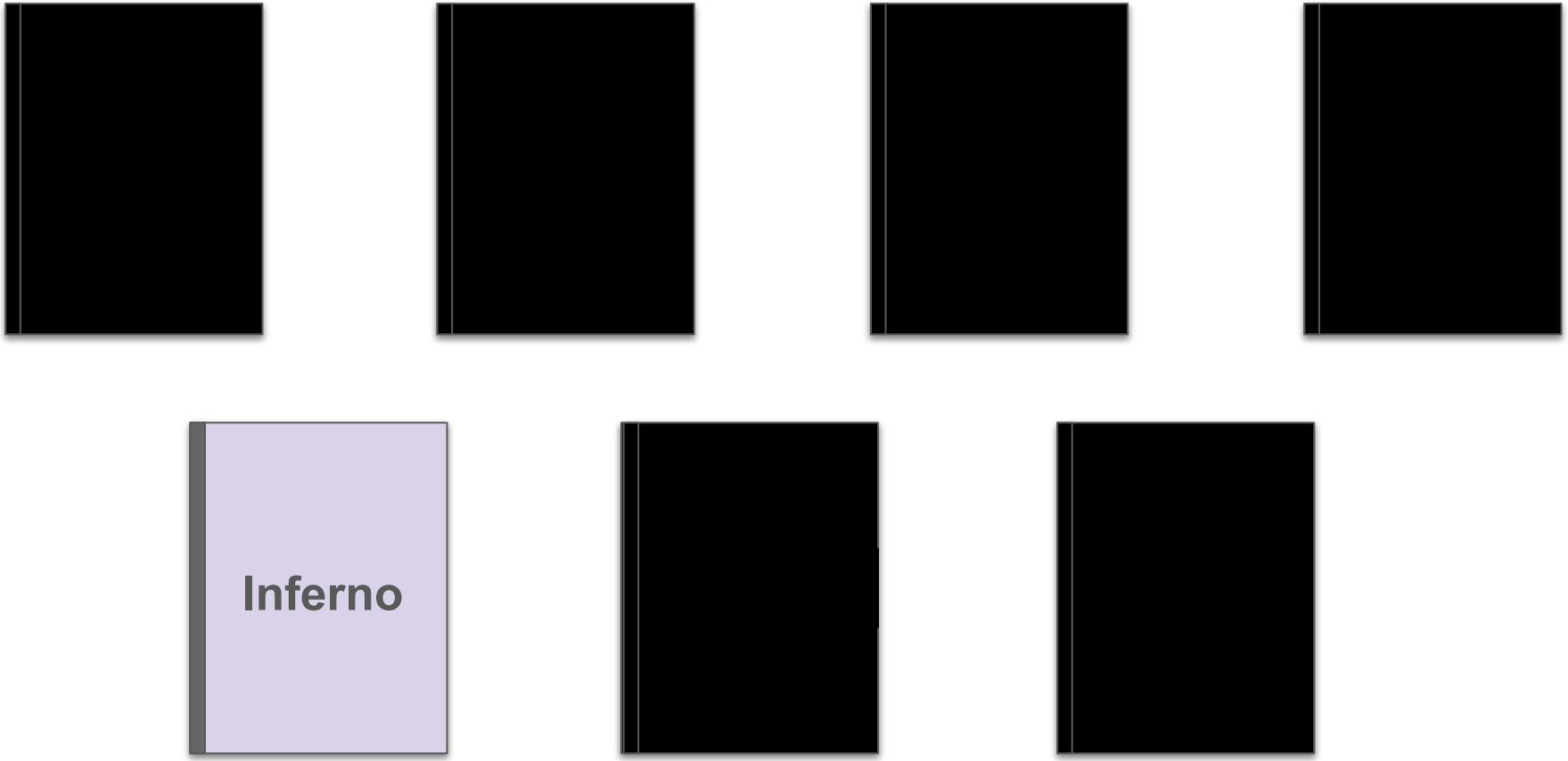
- Searching
- Sorting
- Structs
- Recursion

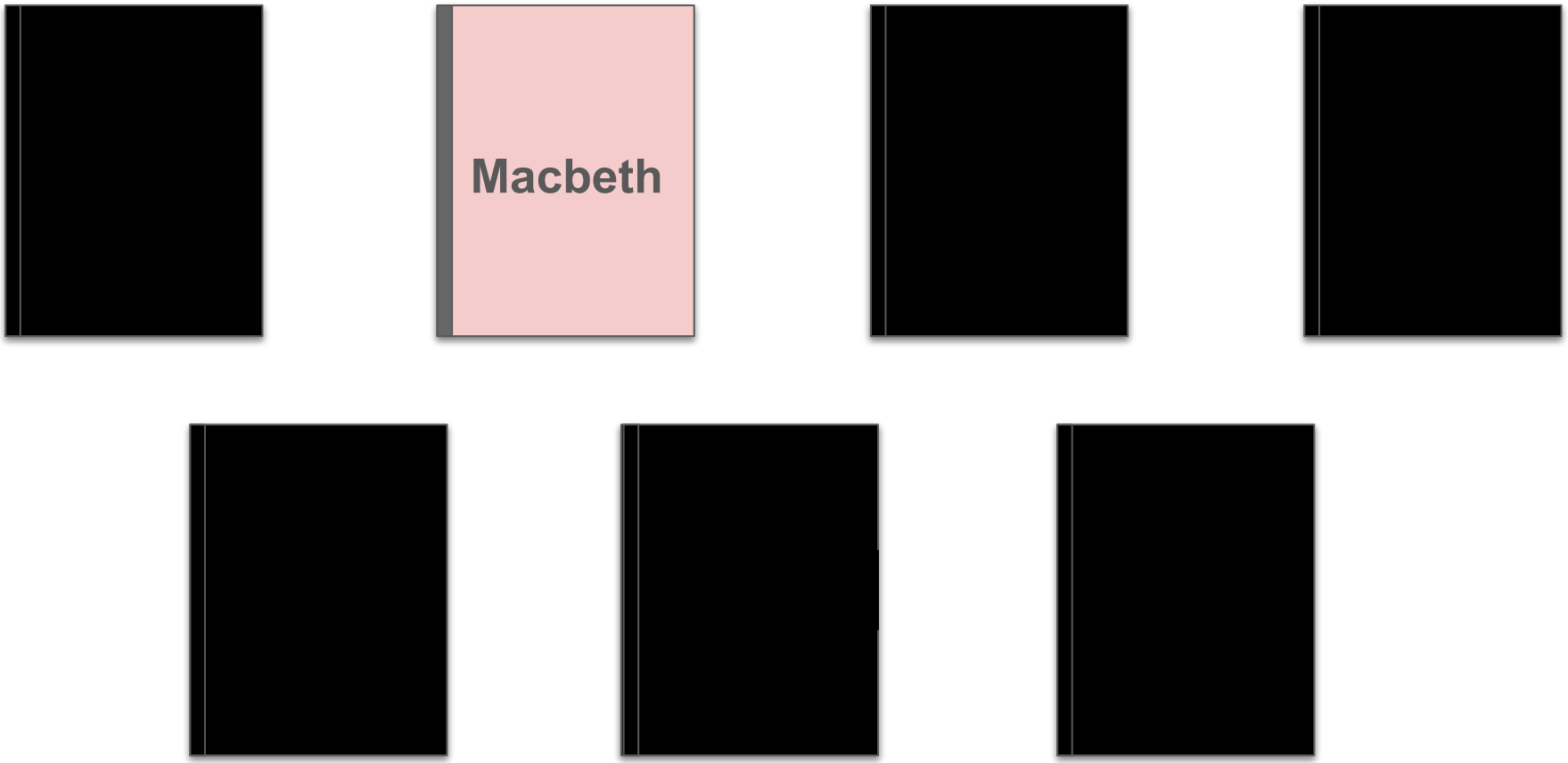
Searching

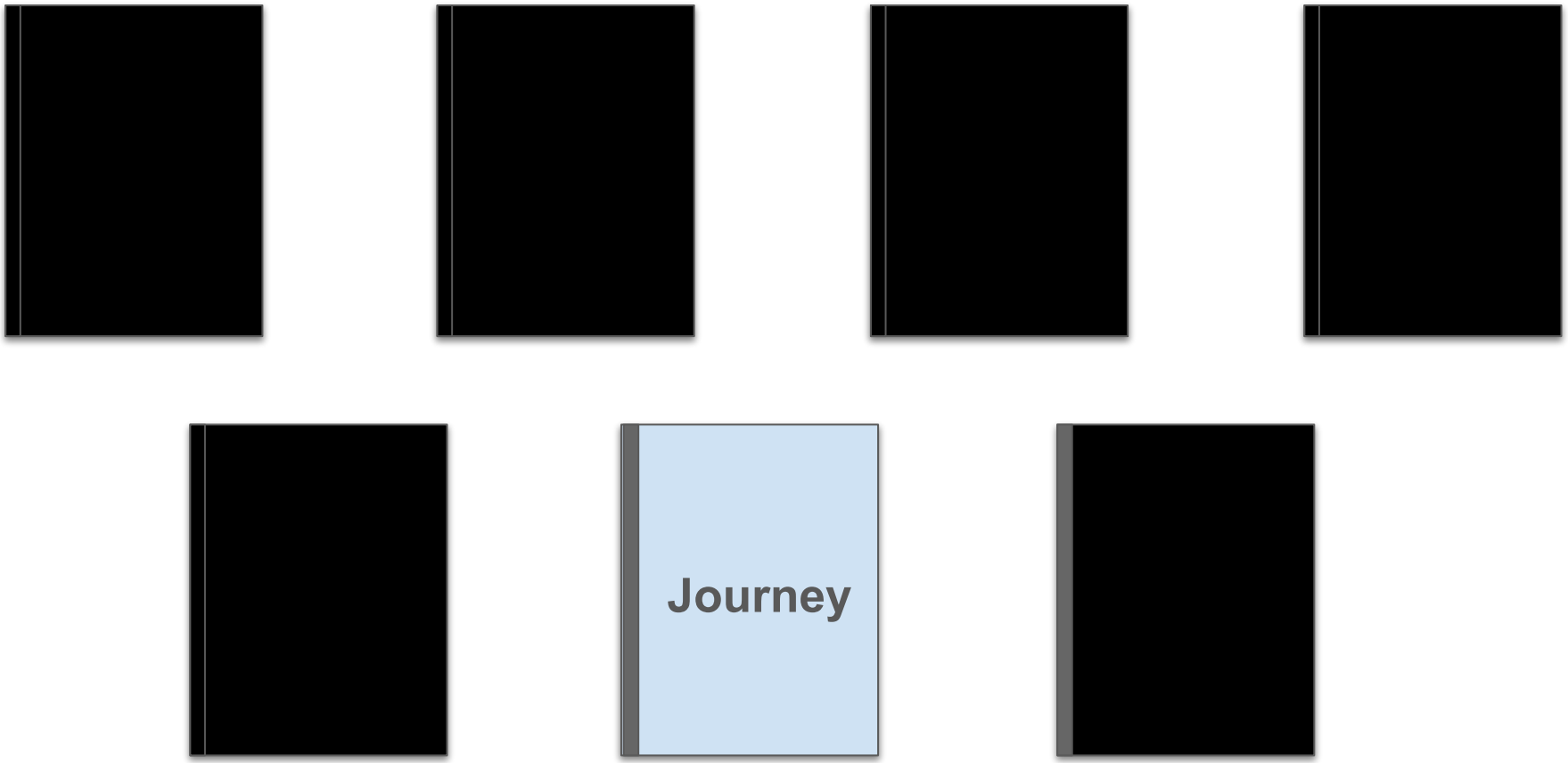

Odyssey
Macbeth
Eclipse
Hamlet
Inferno
Journey
Dune

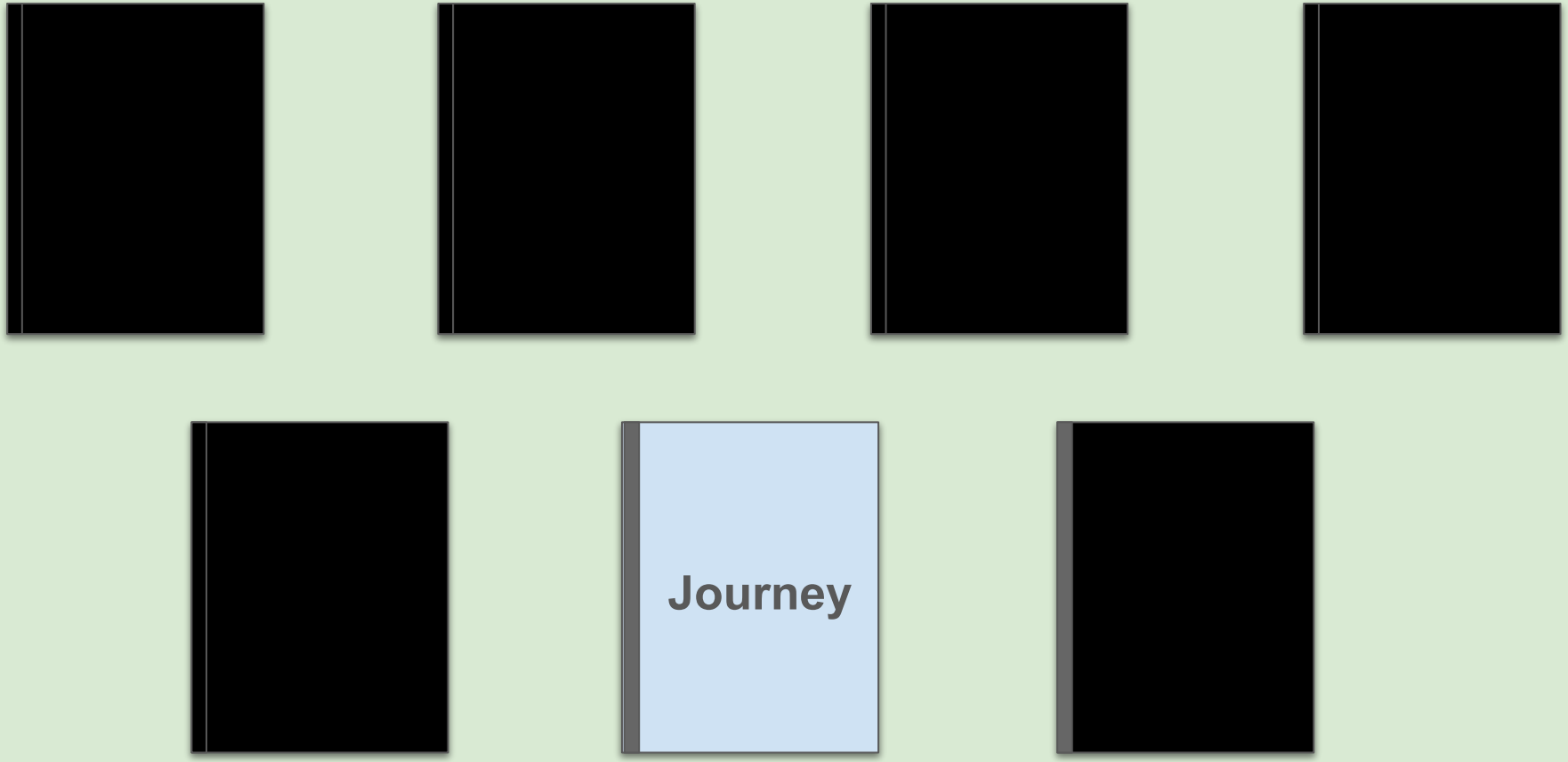
Linear Search


Odyssey


Inferno


Macbeth


Journey


Journey

Linear Search

Binary Search

Dune

Eclipse

Hamlet

Inferno

Journey

Macbeth

Odyssey

Inferno

Macbeth

Journey

Journey

Runtime Analysis

	Linear Search	**Binary Search**
“Worst Case”		
“Best Case”		

Runtime Analysis

	Linear Search	**Binary Search**
“Worst Case”	7	$log_2(7)$
“Best Case”		

Runtime Analysis

	Linear Search	**Binary Search**
“Worst Case”	N	$log_2(N)$
“Best Case”		

Runtime Analysis

	Linear Search	**Binary Search**
“Worst Case”	$O(N)$	$O(\log_2(N))$
“Best Case”		

Runtime Analysis

	Linear Search	**Binary Search**
“Worst Case”	$O(N)$	$O(\log_2(N))$
“Best Case”	1	1

Runtime Analysis

	Linear Search	**Binary Search**
“Worst Case”	$O(N)$	$O(\log_2(N))$
“Best Case”	$\Omega(1)$	$\Omega(1)$

Sorting

Runtime Analysis

Algorithm	***O***	**Ω**
Merge Sort	$O\ (N\log(N))$	$\Omega\ (N\log(N))$
Selection Sort	$O\ (N^2)$	$\Omega\ (N^2)$
Bubble Sort	$O\ (N^2)$	$\Omega\ (N)$

Sort

Algorithm	*reversed50000.txt*	*sorted50000.txt*
sort1		
sort2		
sort3		

Structs

```
typedef struct
{
   string name;
   int votes;
}
candidate;
```

```
typedef struct
{
   string name;
   int votes;
}
candidate;
```

```
typedef struct
{
   string name;
   int votes;
}
candidate;
```

```
typedef struct
{
   string name;
   int votes;
}
candidate;
```

```
candidate president;
```

```
candidate president;
```

type

```
candidate president;
```

variable name

```
candidate president;

president.name = "Alice";
president.votes = 10;
```

What if we had multiple candidates?

name	Alice	Bob	Charlie
votes	2	1	3

name	Alice	Bob	Charlie
votes	2	1	3

```
candidates[0];
```

name	Alice	Bob	Charlie
votes	2	1	3

```
candidates[0].name;
```

name	Alice	Bob	Charlie
votes	2	1	3

```
candidates[0].votes;
```

Structs Exercise

- Create an array of three candidates
- Populate an array using user's input
- Search the array to find the most votes awarded to any single candidate

Recursion

Factorial

Factorial

$$1! = 1$$

$$2! = 1 * 2$$

$$3! = 1 * 2 * 3$$

$$4! = 1 * 2 * 3 * 4$$

Factorial

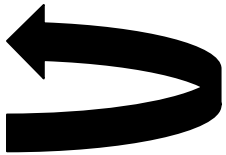
```
4! = 4 * 3!

3! = 3 * 2!

2! = 2 * 1!

1! = 1
```

Factorial

```
f(4)
  4 * f(3)
    3 * f(2)
      2 * f(1)
        1 * f(0)
```

Factorial

```
f(4)
  4 * f(3)
    3 * f(2)
      2 * f(1)
        1 * f(0)
```



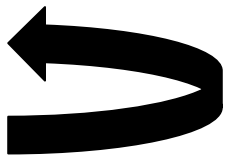
1

Factorial

```
f(4)
  4 * f(3)
    3 * f(2)
      2 * 1
```

Factorial

```
f(4)
  4 * f(3)
    3 * f(2)
      2 * 1
```



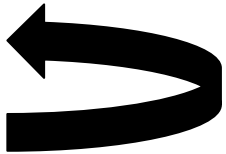
2

Factorial

```
f(4)
  4 * f(3)
    3 * 2
```

Factorial

```
f(4)
  4 * f(3)
    3 * 2
```



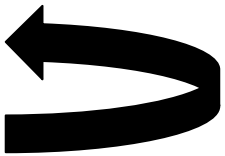
6

Factorial

```
f(4)
  4 * 6
```

Factorial

```
f(4)
  4 * 6
```



24

Factorial

24

This is CS50

Week 3