CS 103000
Prof. Madeline Blount

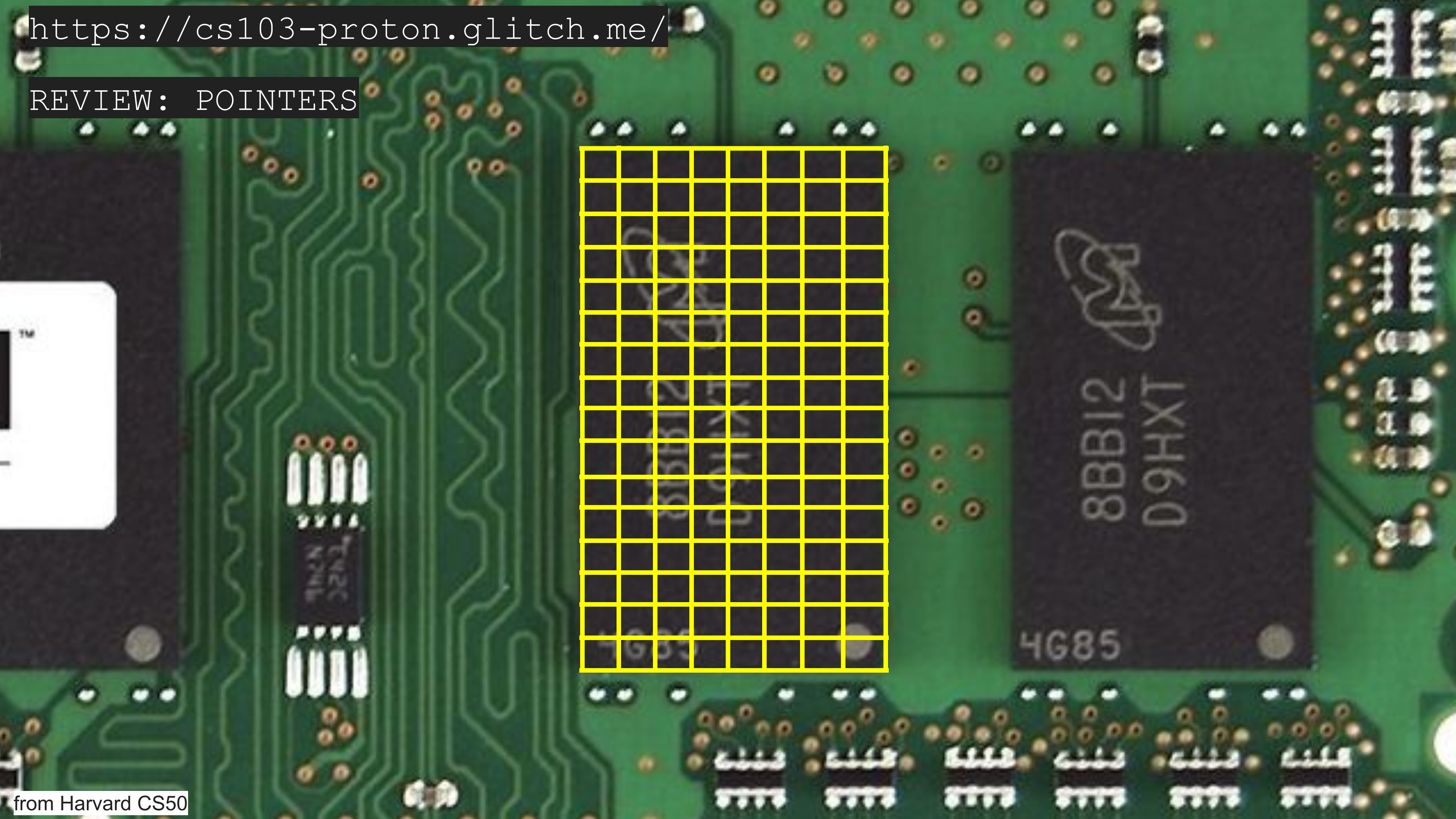
Week 13:
ALGORITHMS

attendance link:

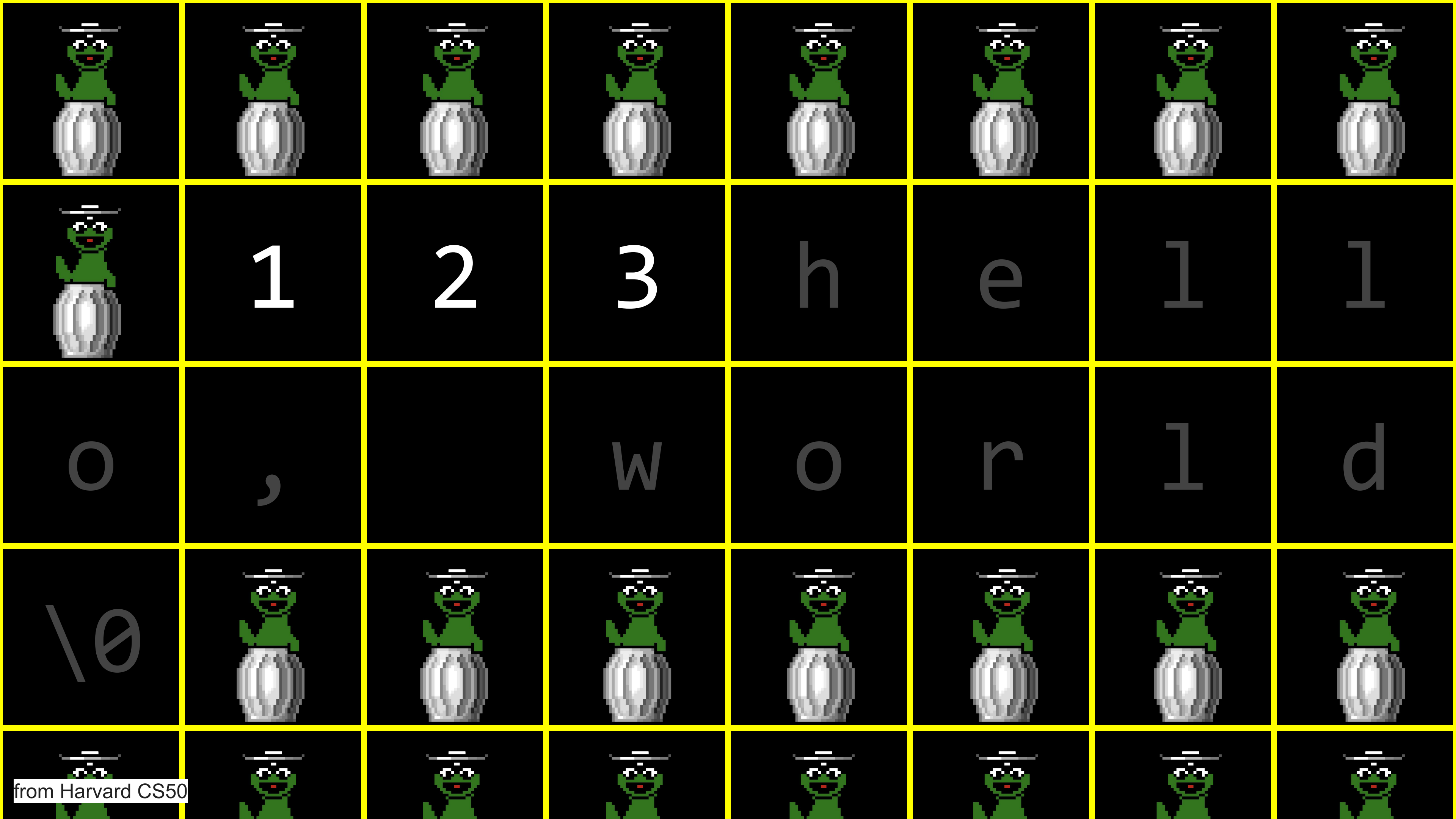
https://cs103-proton.glitch.me/

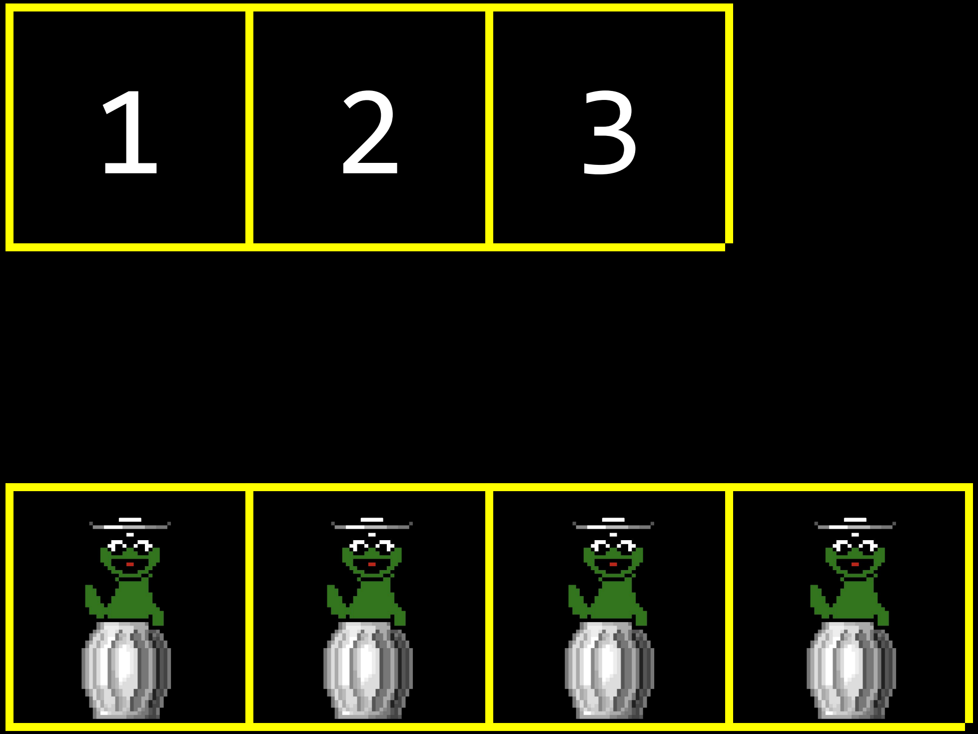
Dall-E 2: cats learning C++ in the forest on ‘90’s technology https://cs103-proton.glitch.me/

REVIEW: POINTERS

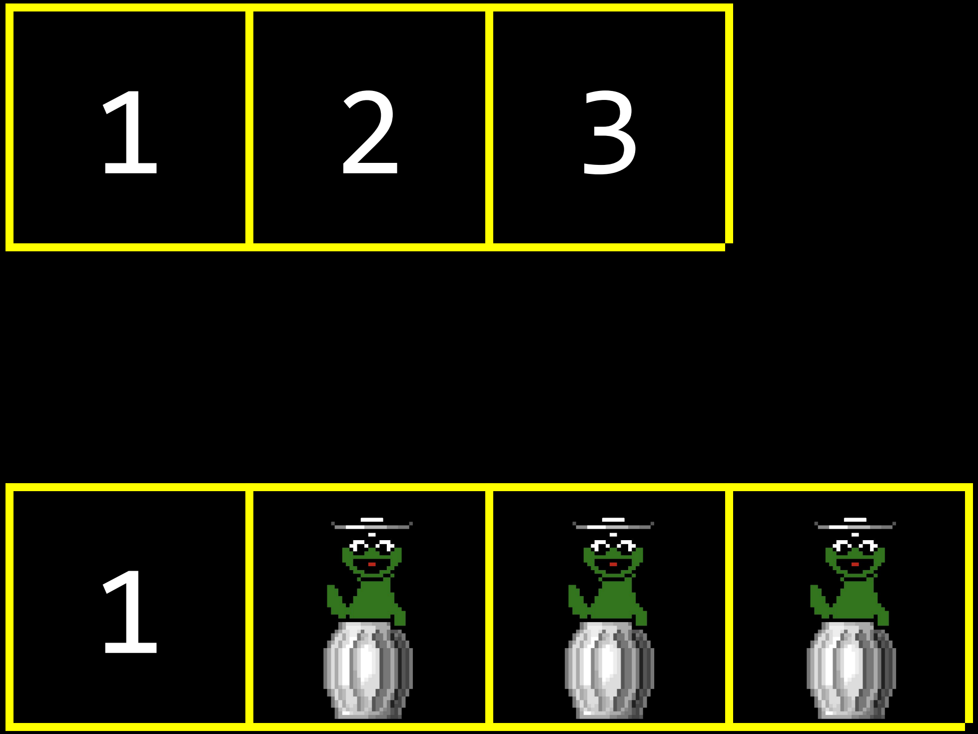


from Harvard CS50

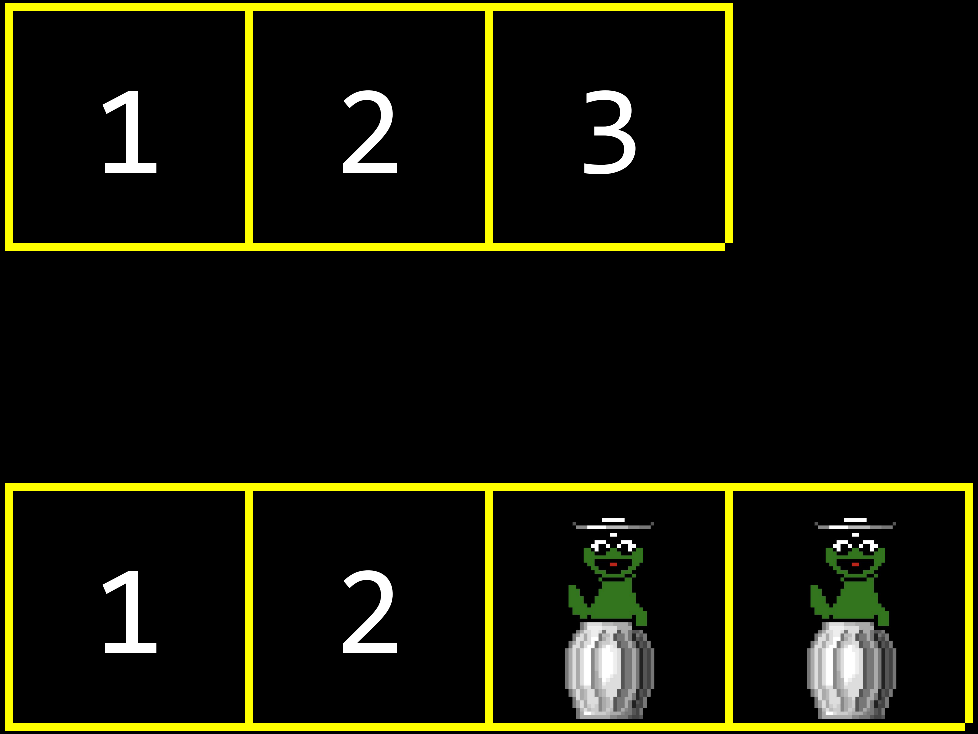

1
2
3
h
e
l
l
o
,
w
o
r
l
d
\0
from Harvard CS50



from Harvard CS50



from Harvard CS50



from Harvard CS50

1	2	3

1	2	3	

from Harvard CS50

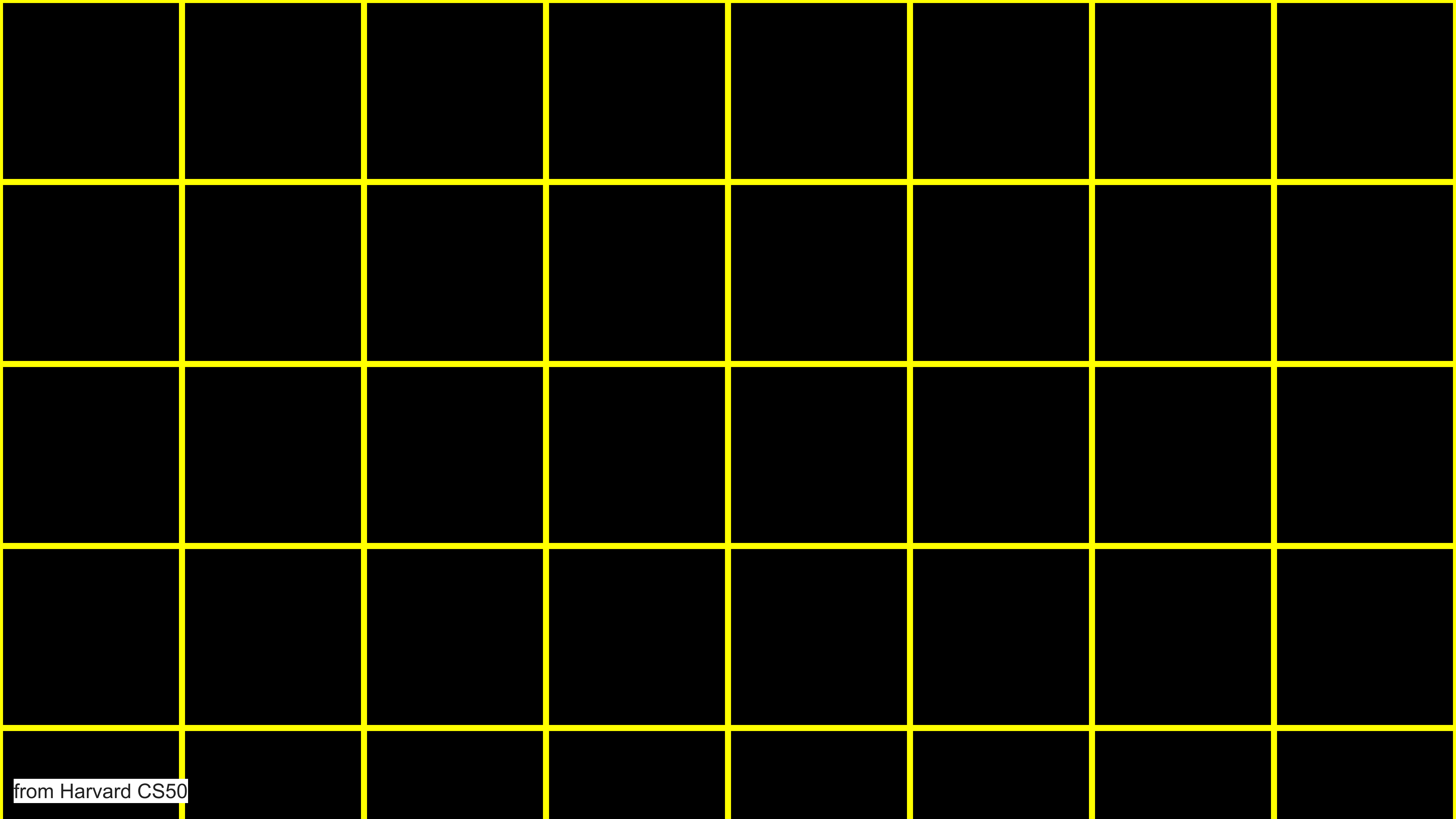
1	2	3

1	2	3	4

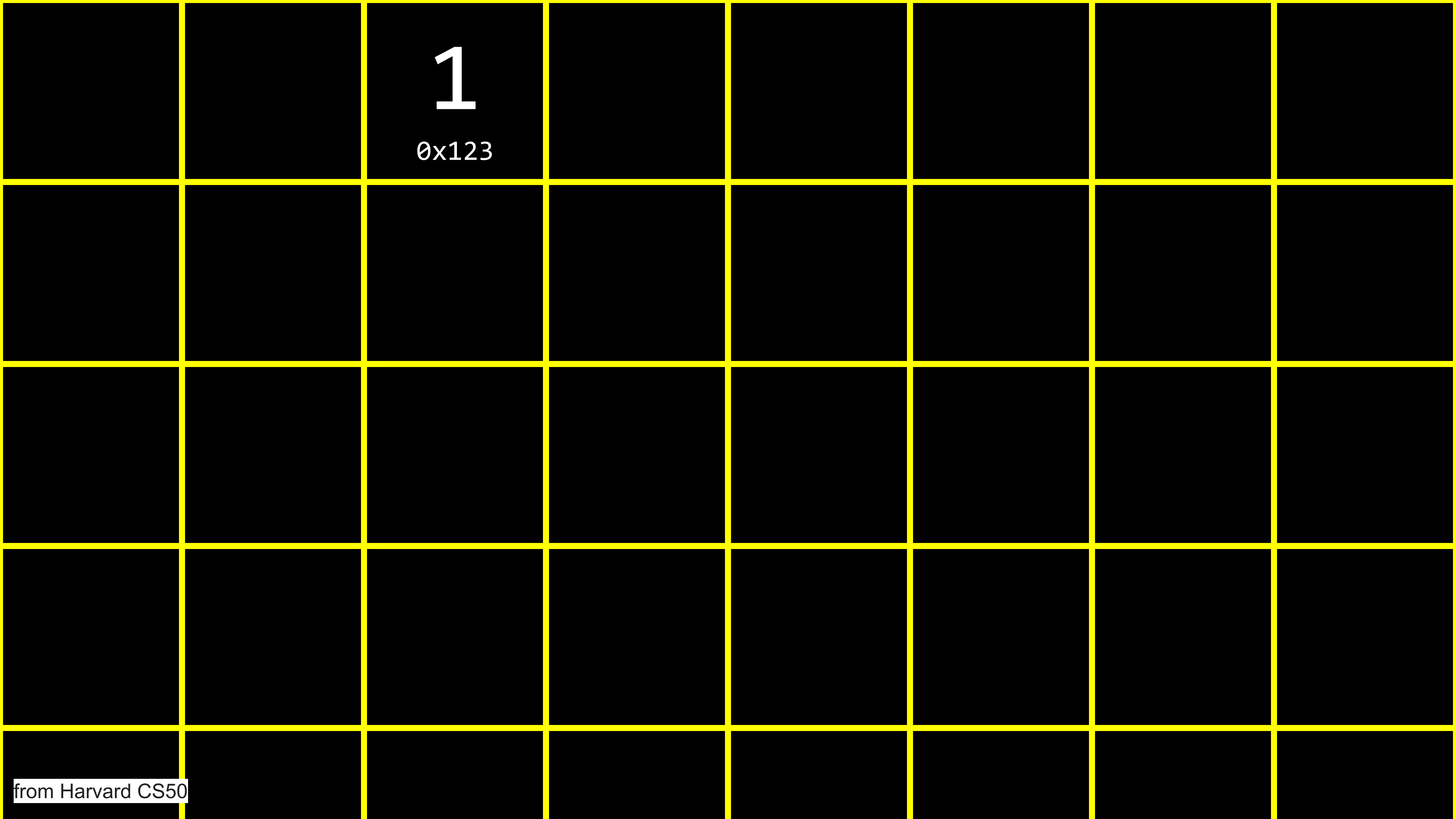
from Harvard CS50

1	2	3	4

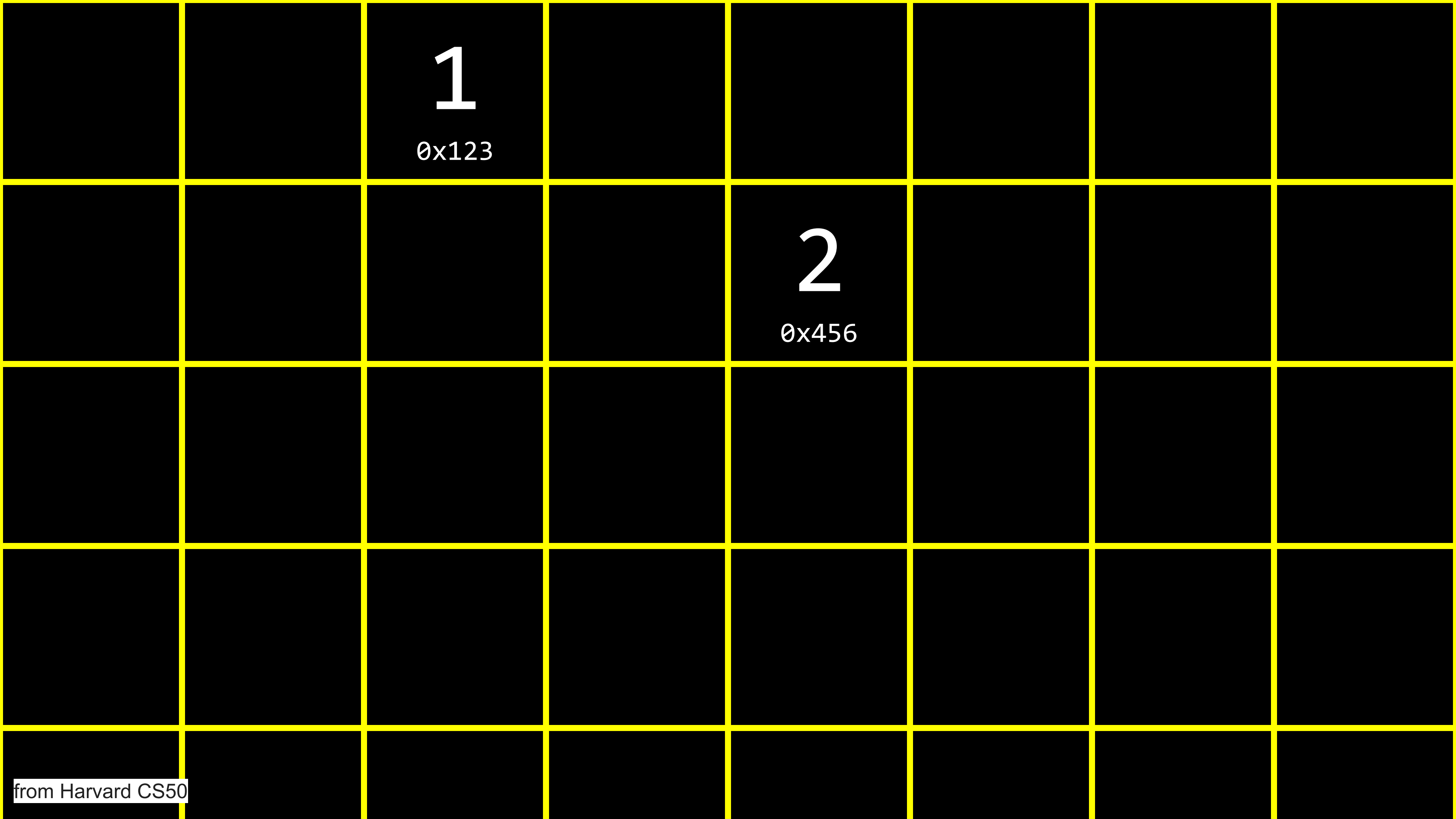
linked lists

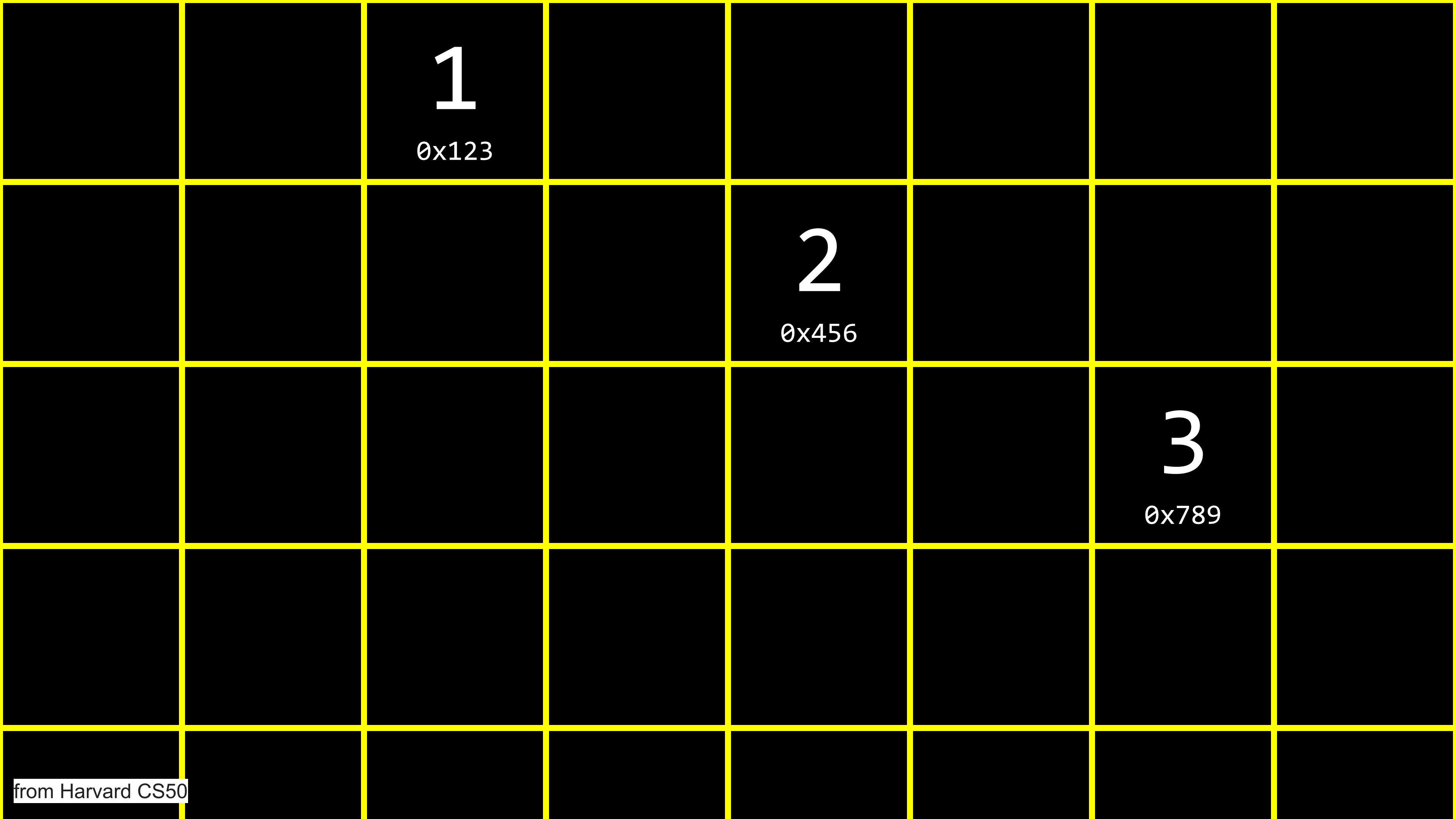


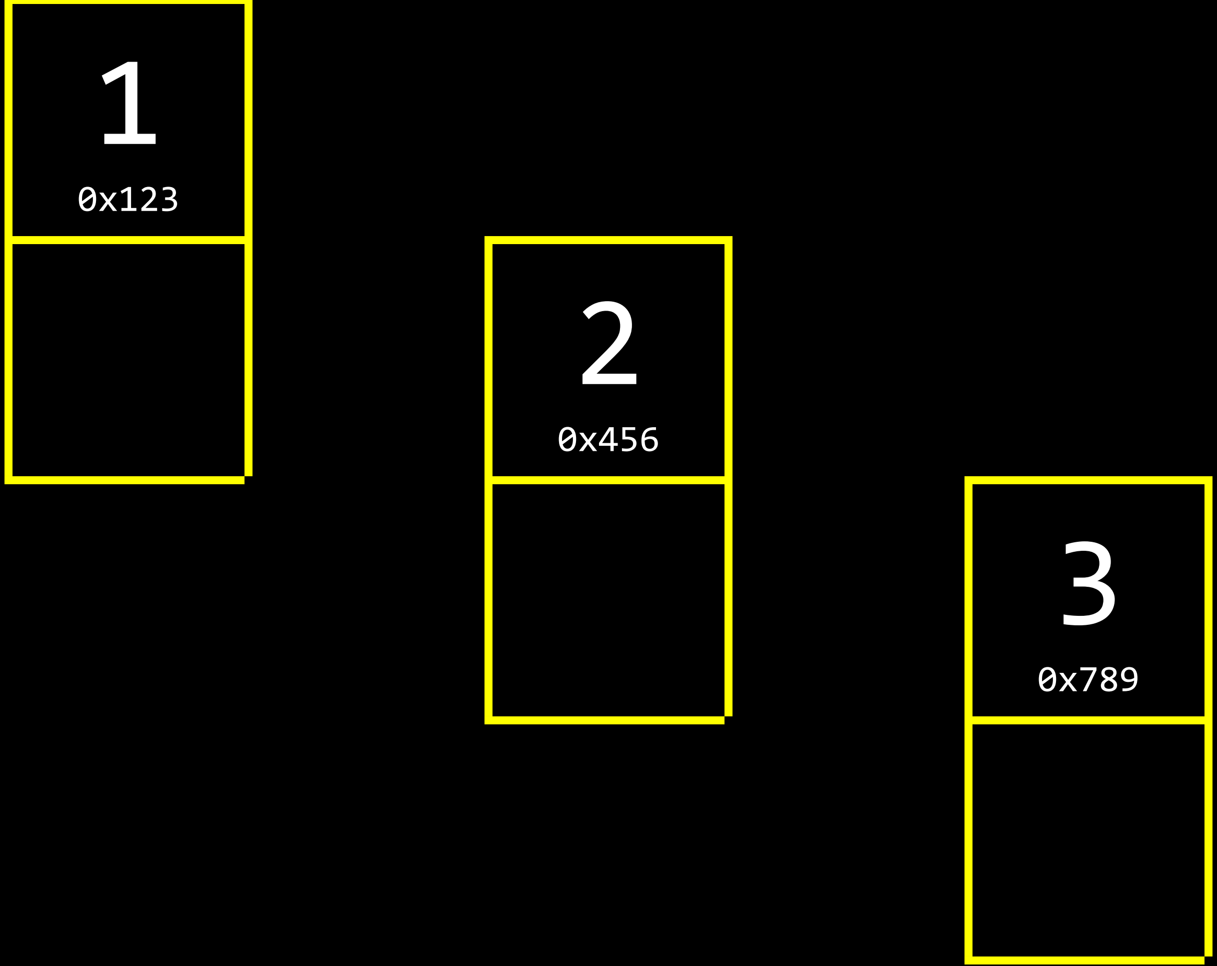
from Harvard CS50



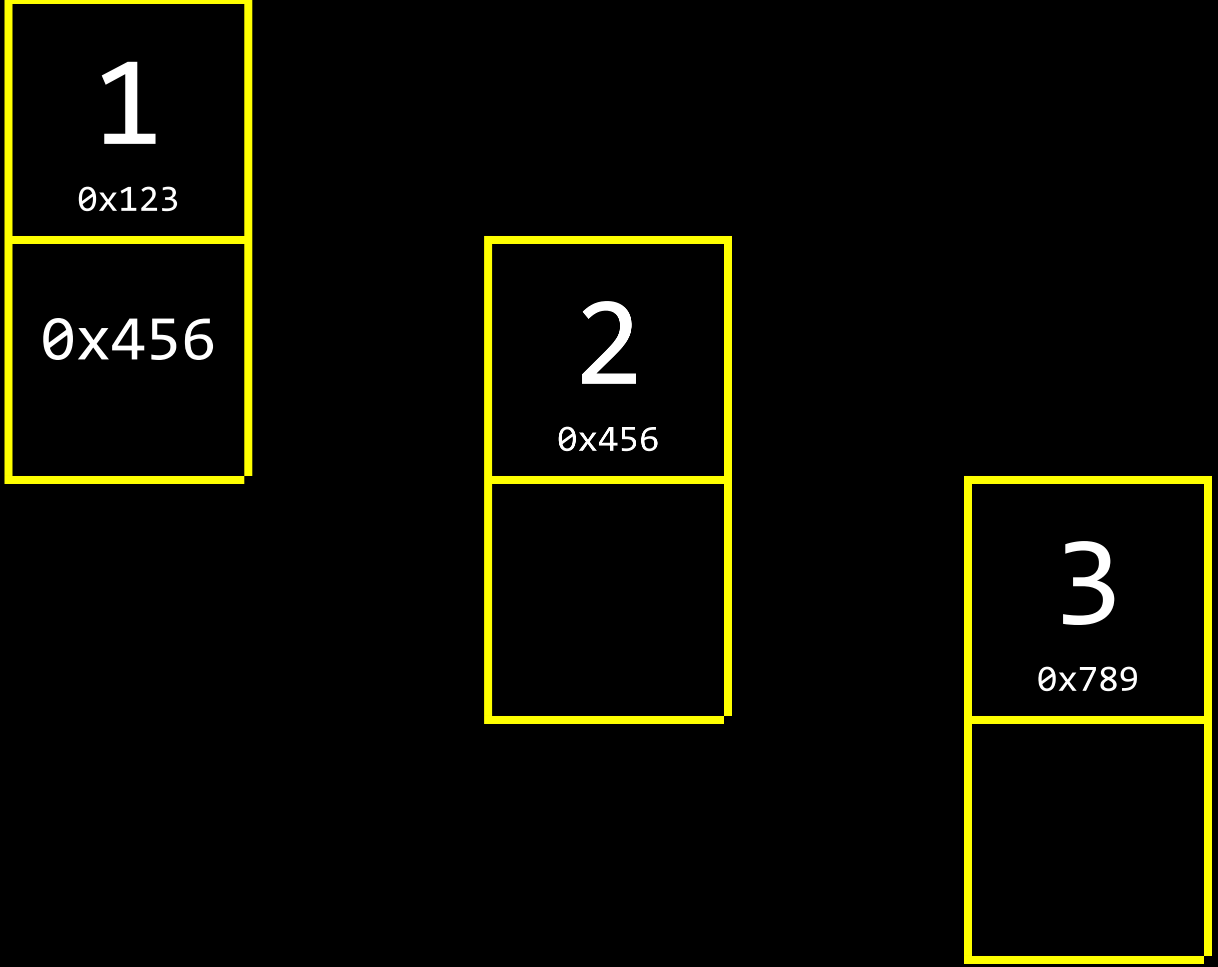
from Harvard CS50


1
0x123
2
0x456
from Harvard CS50

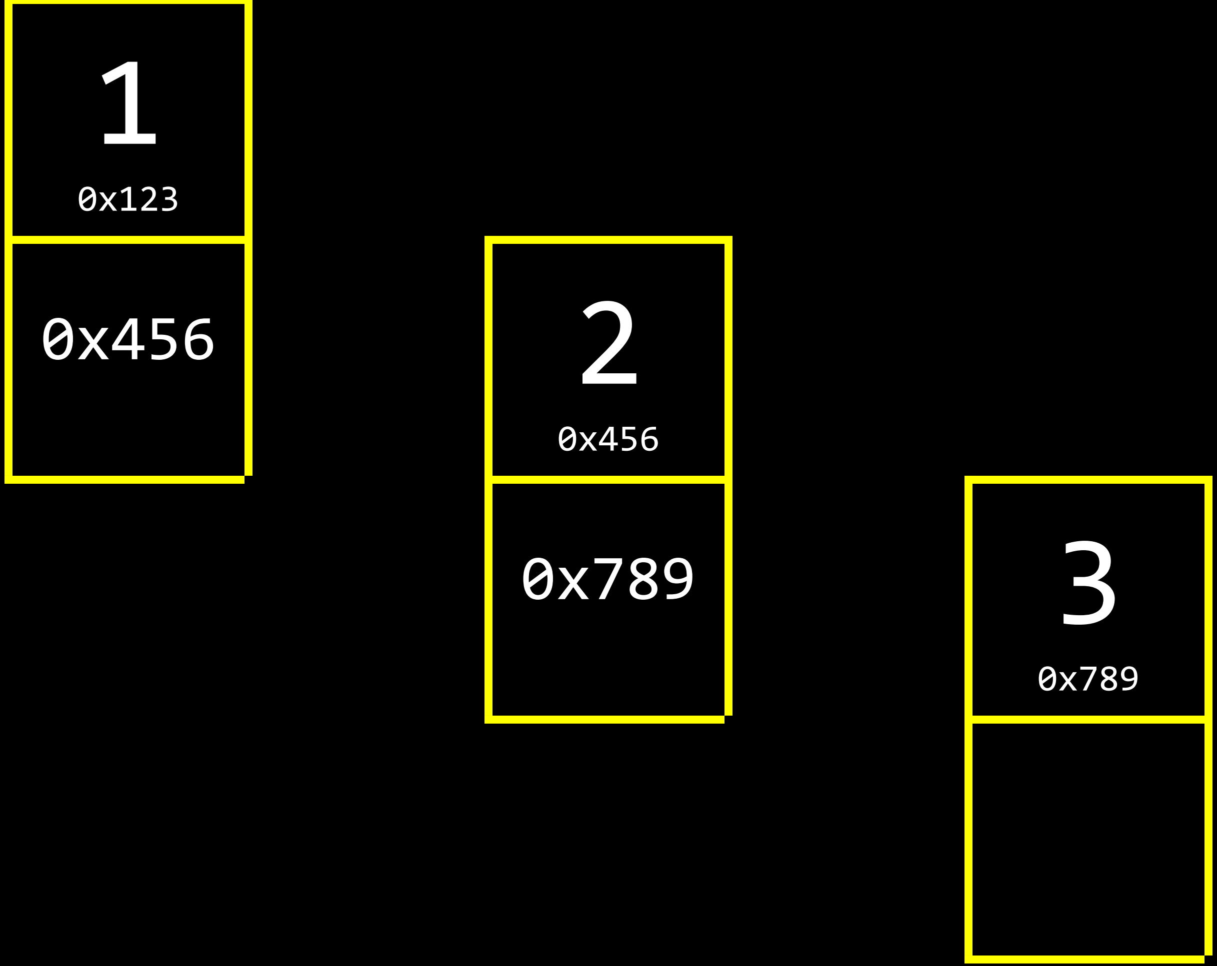

1
0x123
2
0x456
3
0x789
from Harvard CS50



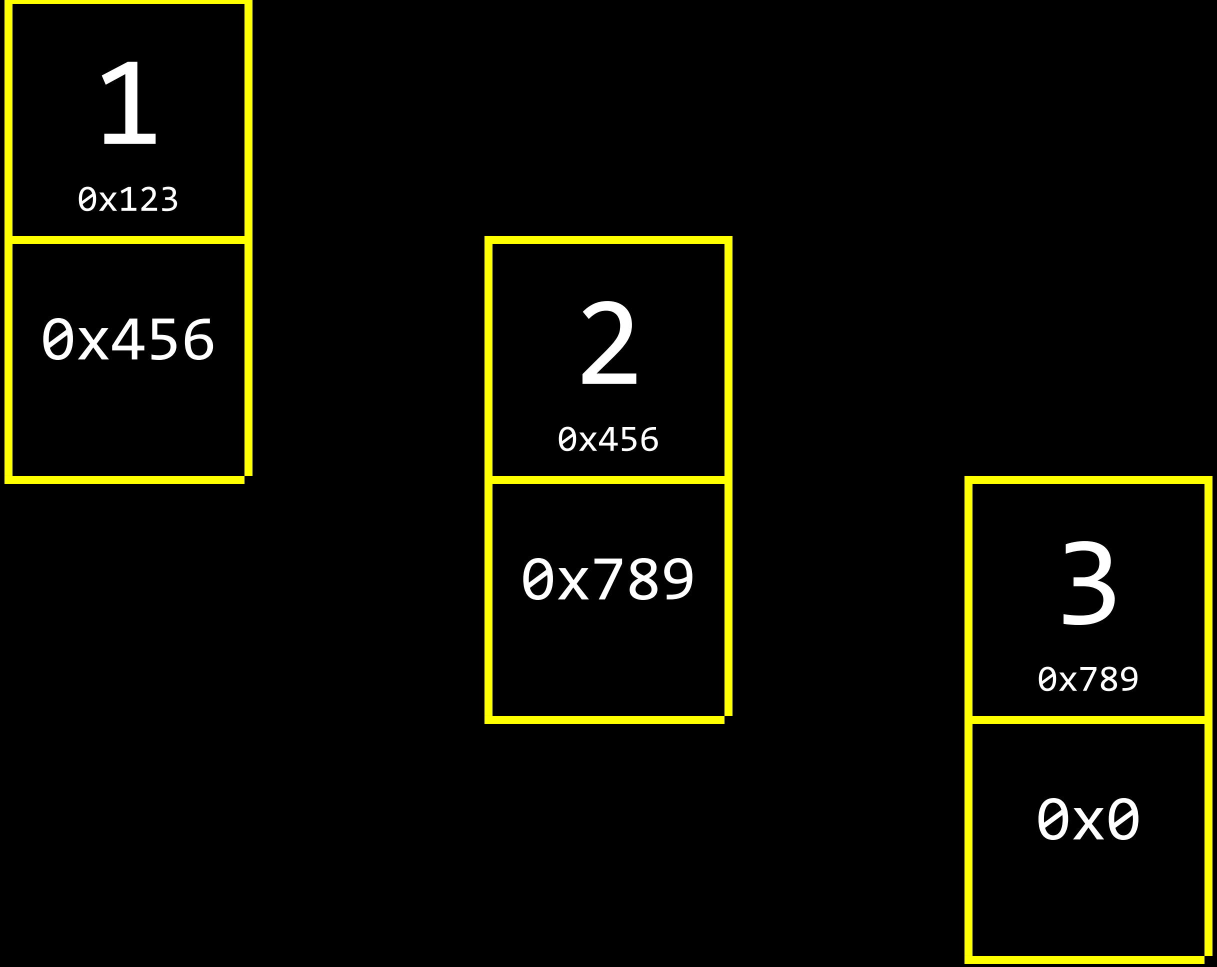
from Harvard CS50



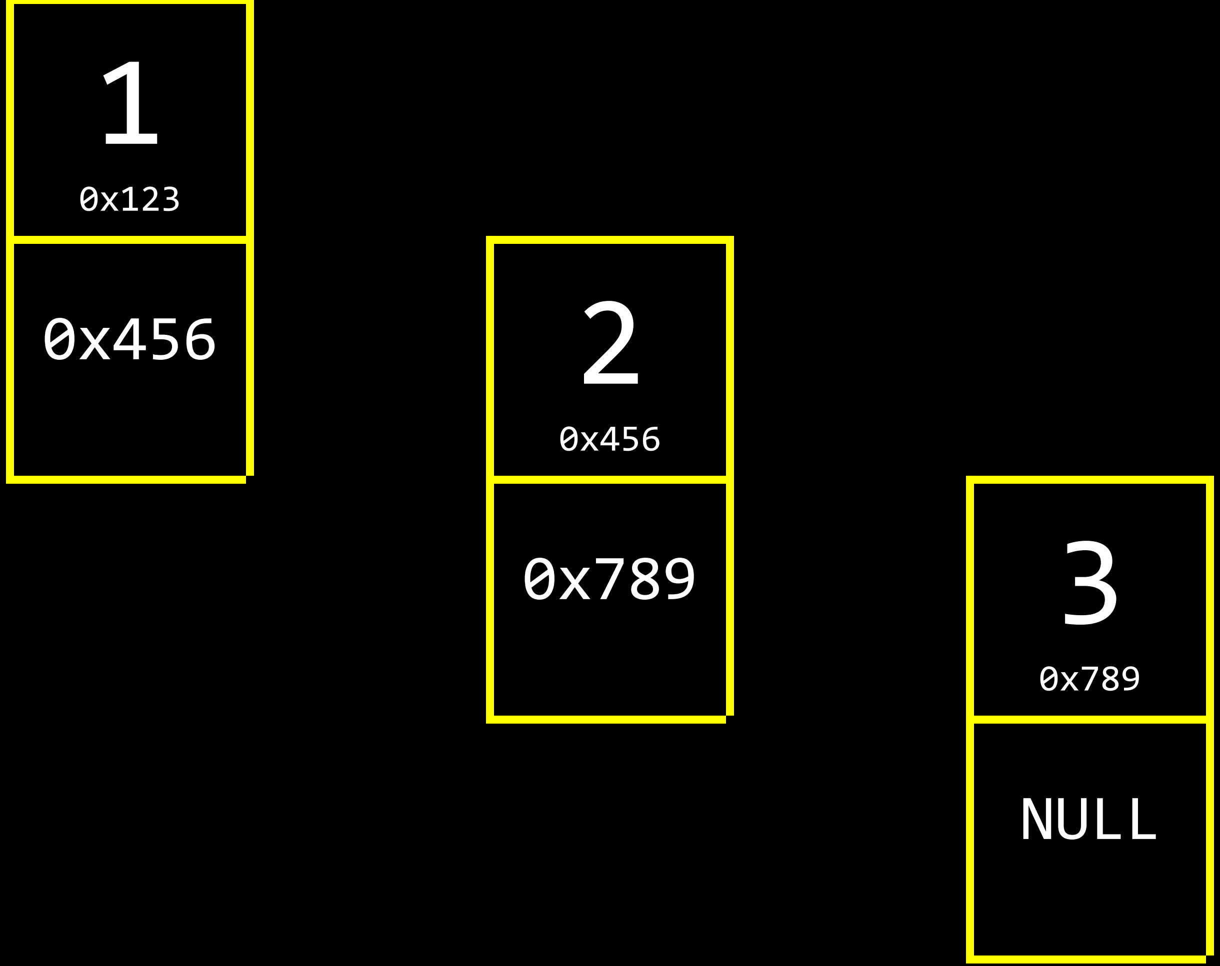
from Harvard CS50



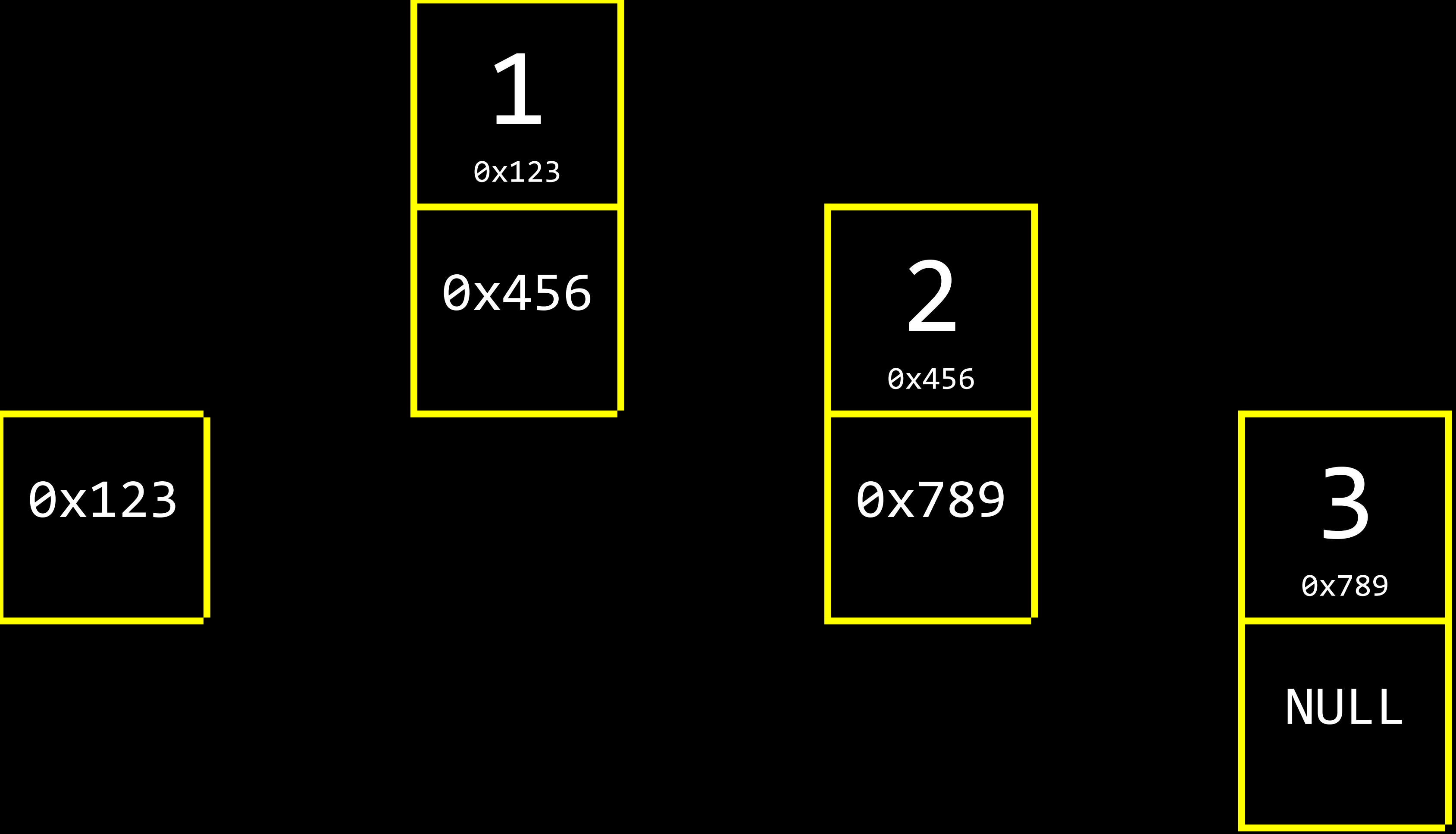
from Harvard CS50



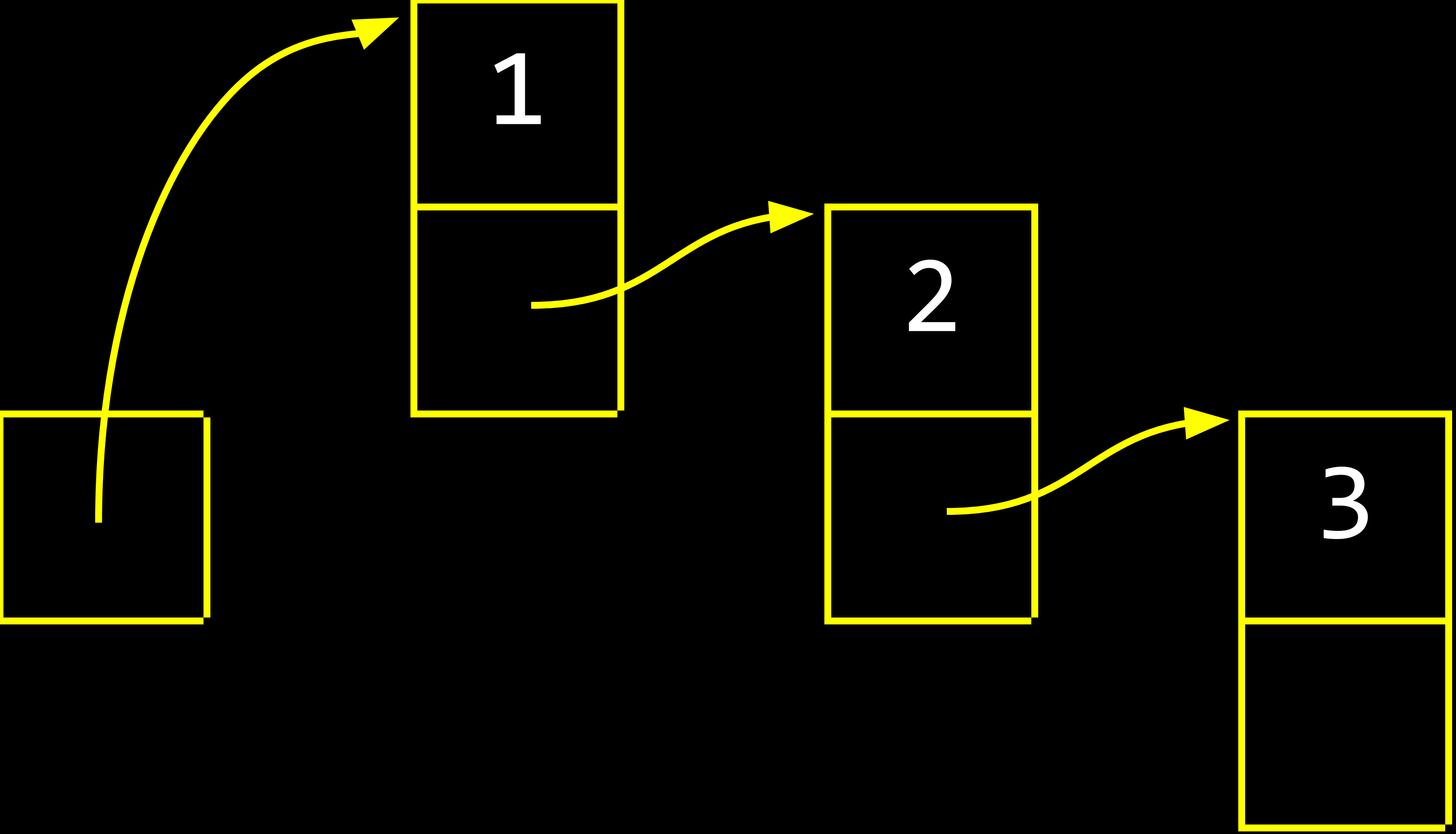
from Harvard CS50



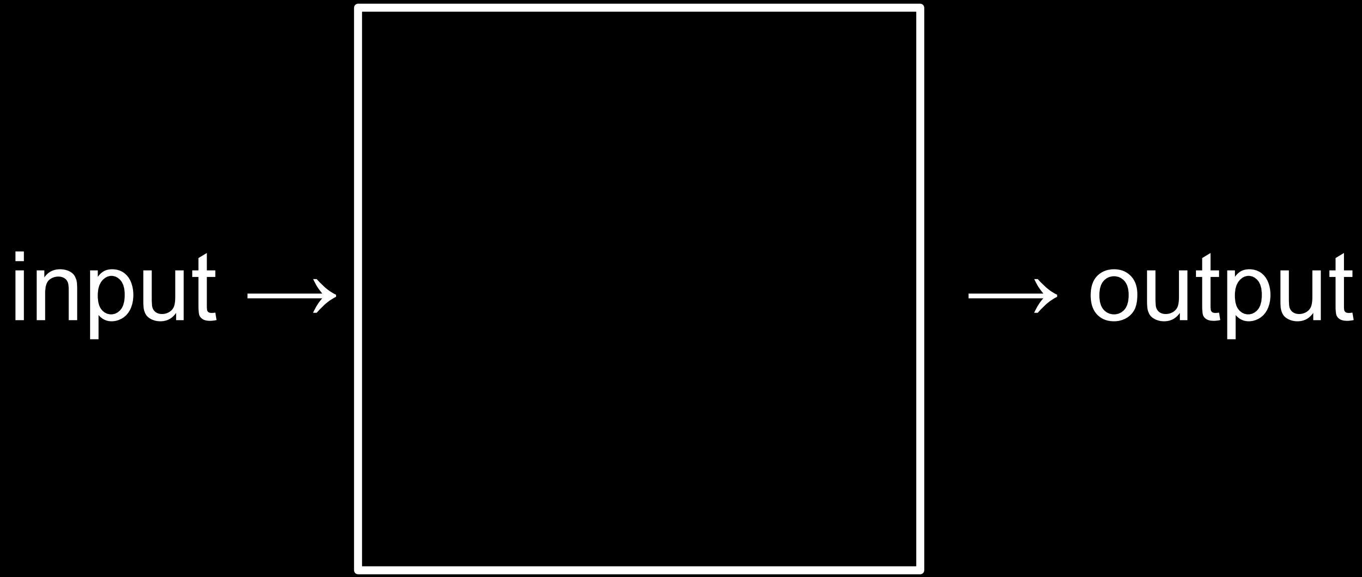
from Harvard CS50



from Harvard CS50



from Harvard CS50

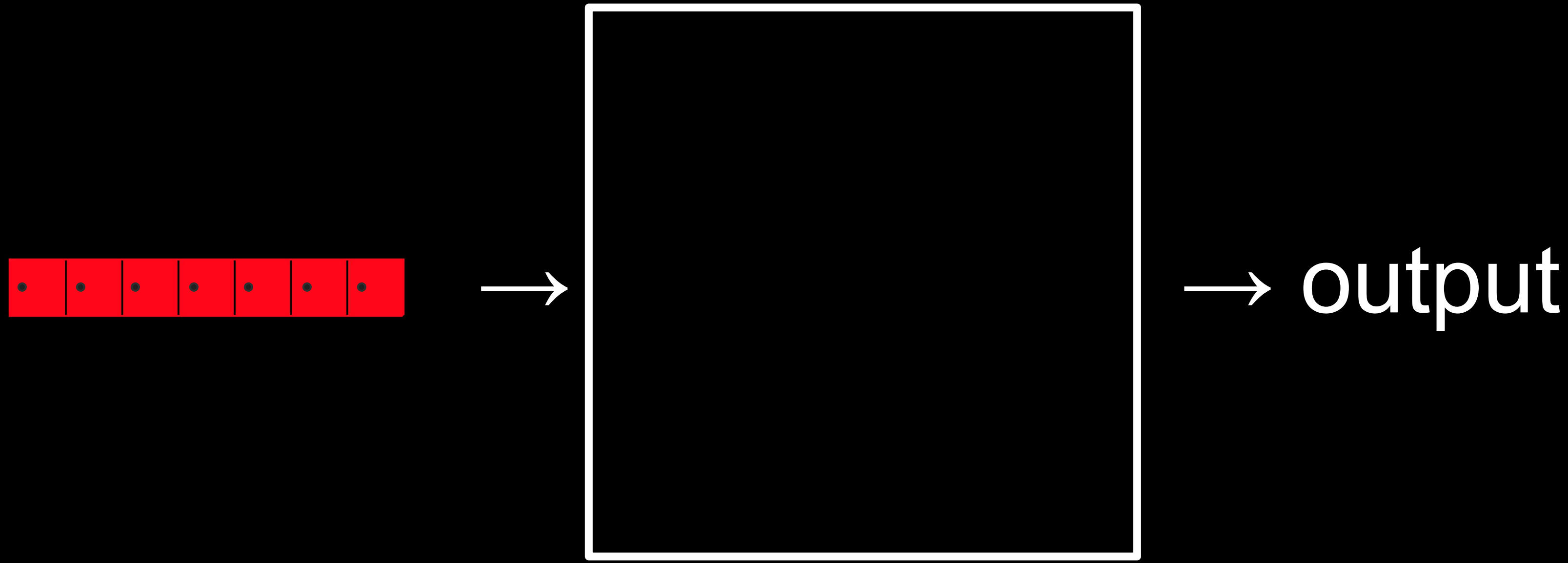


from Harvard CS50

algorithm

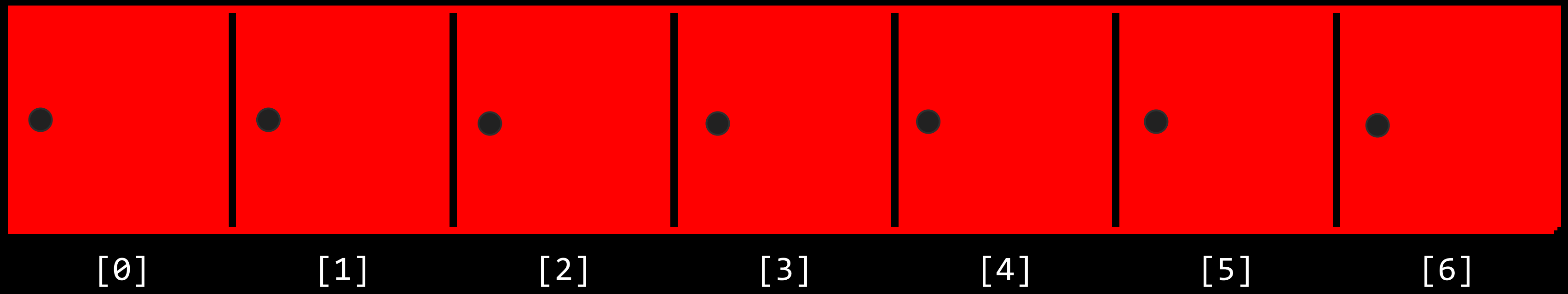
from Harvard CS50

SEARCH

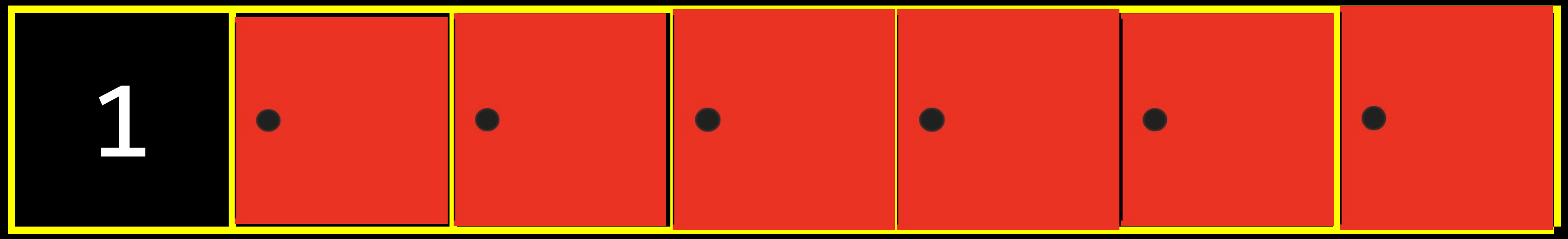


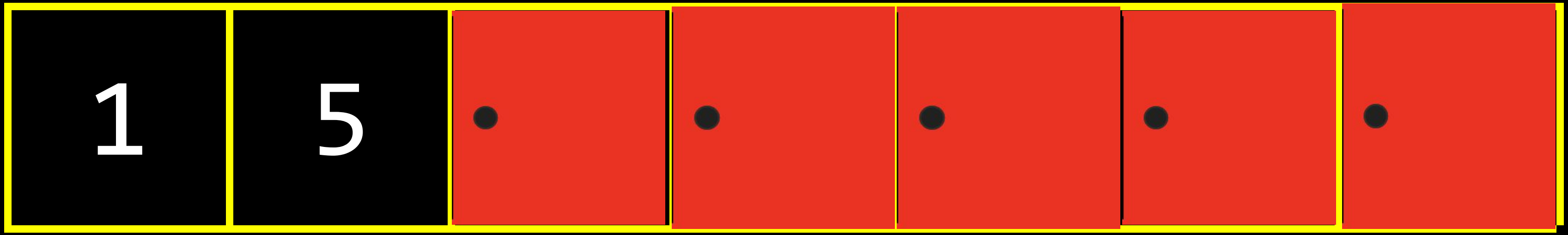
from Harvard CS50

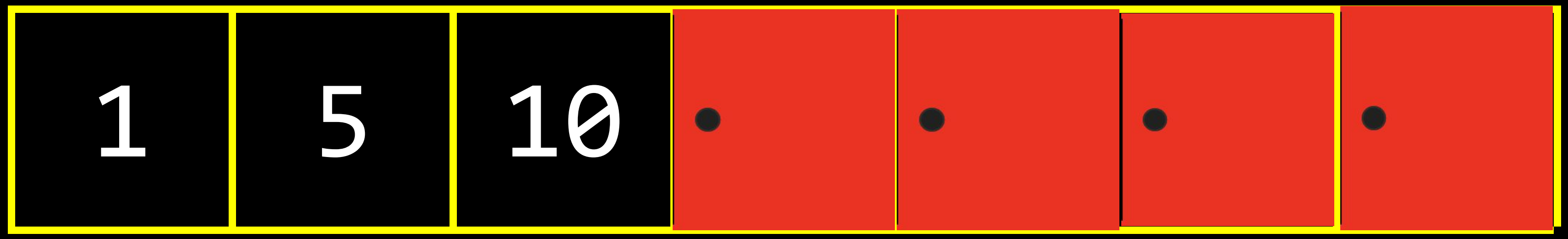
SEARCH

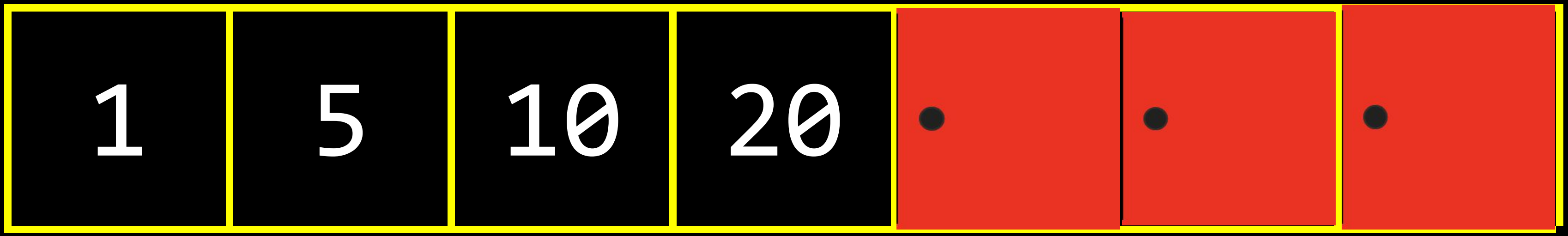


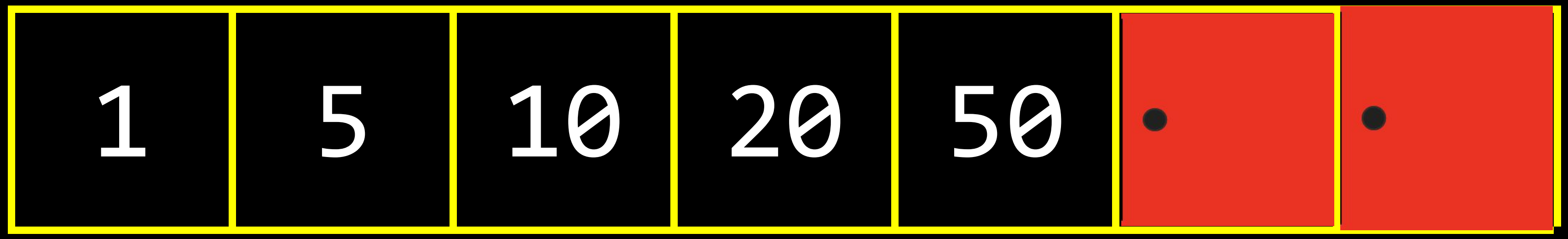
from Harvard CS50

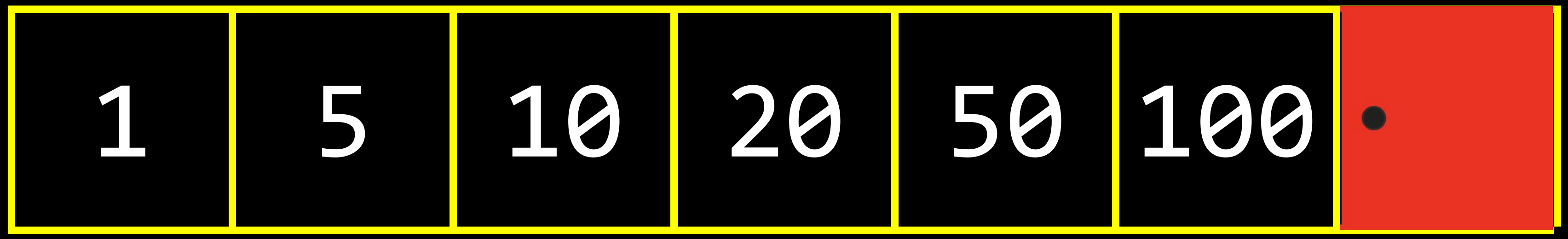

1


1
5

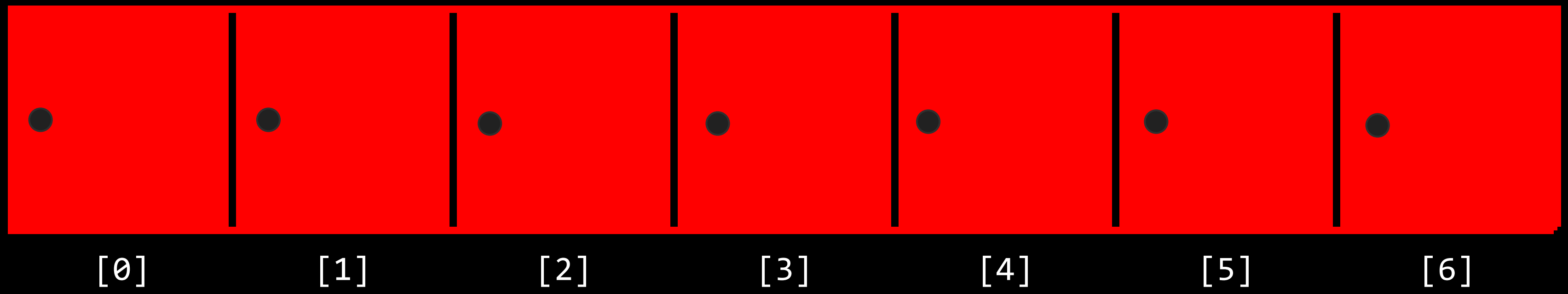

1
5
10

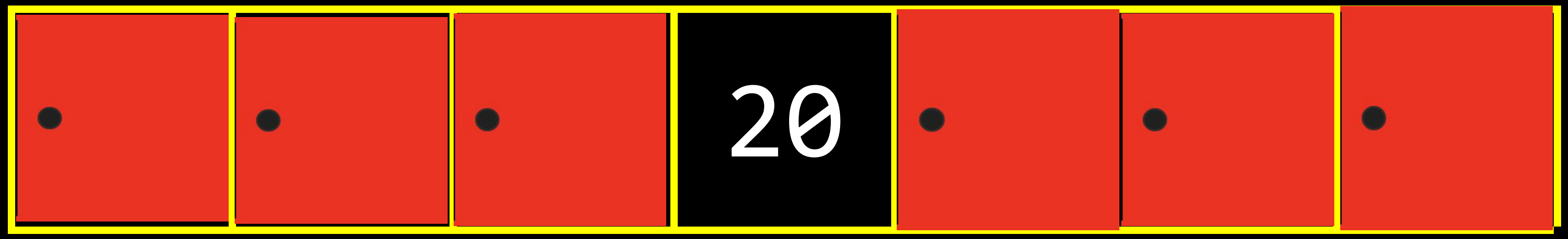

1
5
10
20

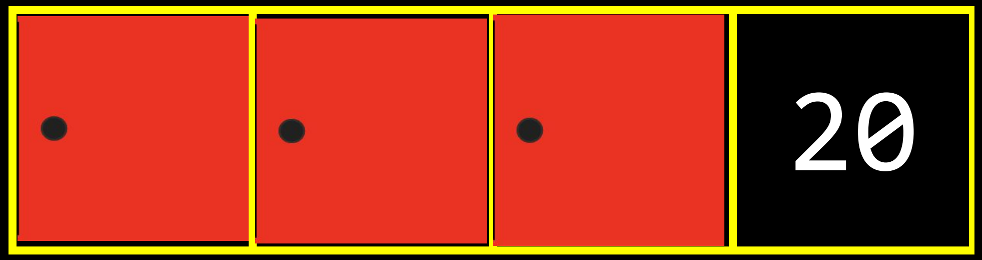

1
5
10
20
50

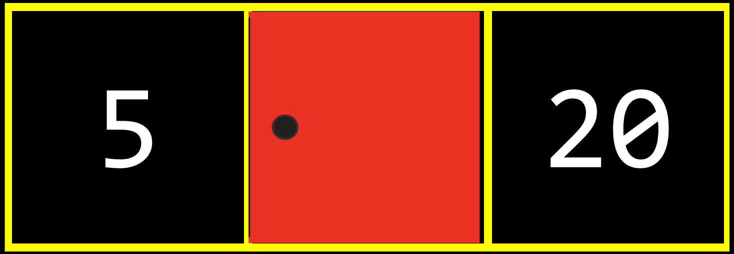

1
5
10
20
50
100

1	5	10	20	50	100	500


[0]
[1]
[2]
[3]
[4]
[5]
[6]


20


20


5
20

5	10	20

1	5	10	20	50	100	500

from Harvard CS50

<u>LINEAR SEARCH: “brute force”</u>

- check every element
- move onto next element

<u>BINARY SEARCH: “divide and conquer”</u>

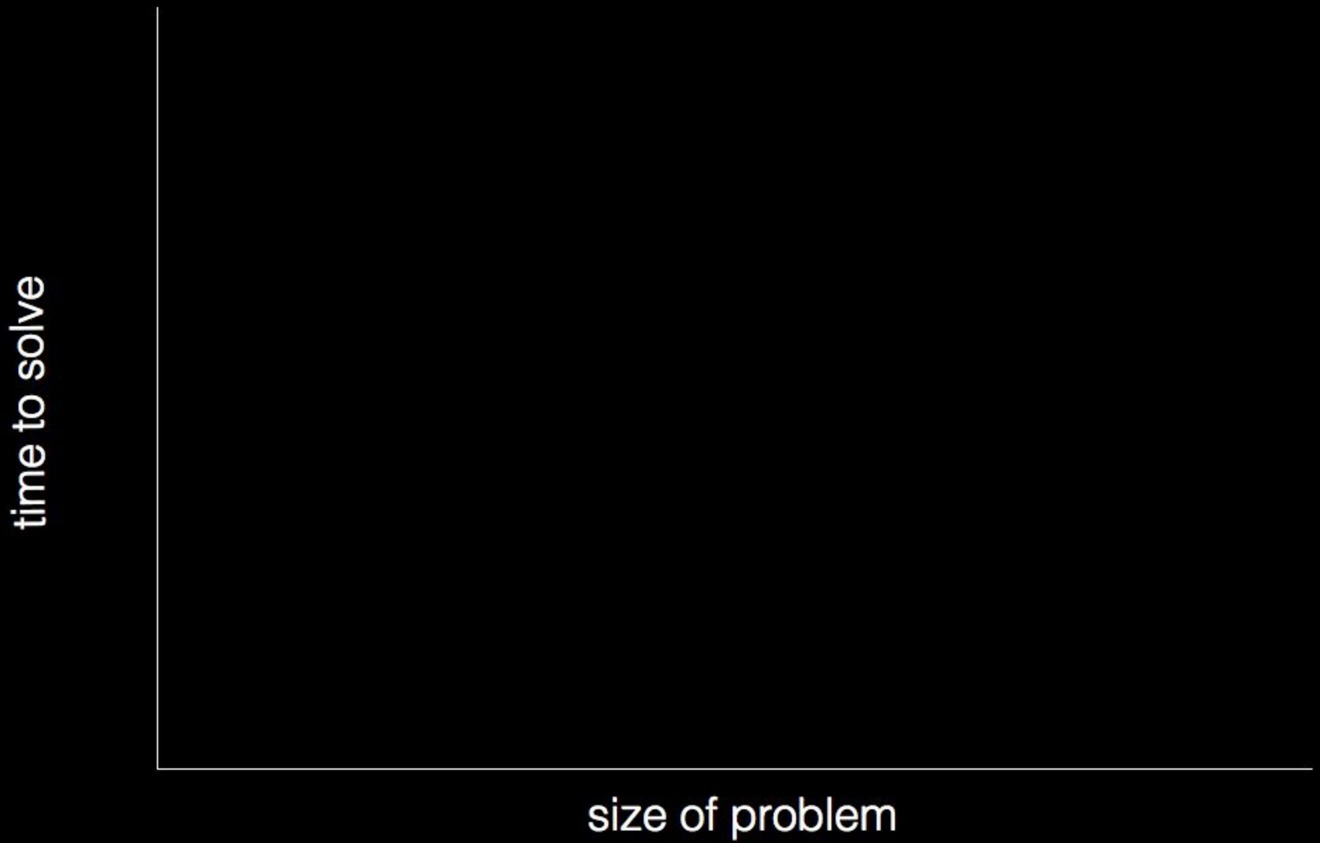
- sorted list
- check middle of list
- if search term > middle, **binary search** to the right
- if search term < middle, **binary search** to the left

recursion: when a process invokes itself

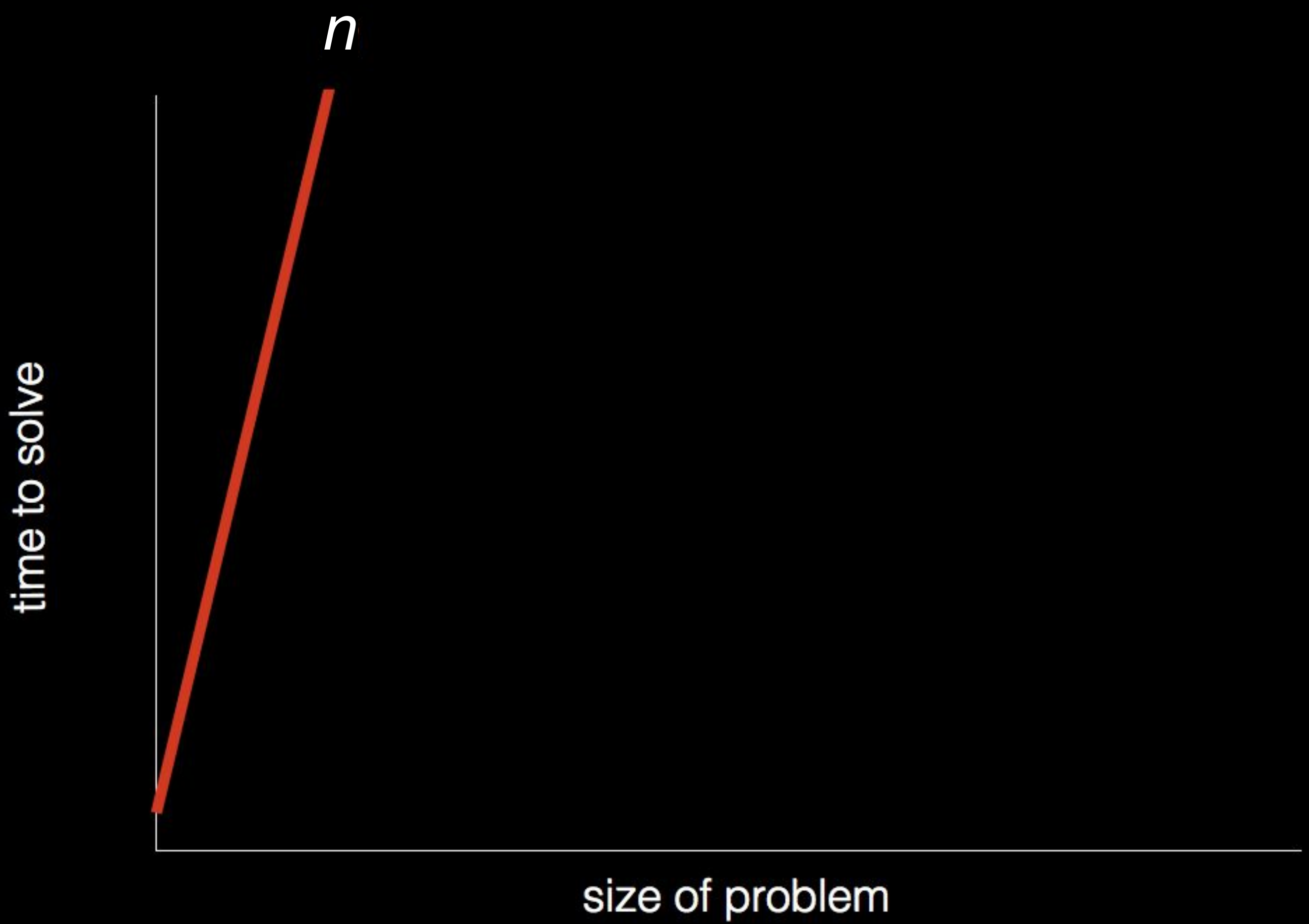
🔎 **BINARY SEARCH:**

- sorted list

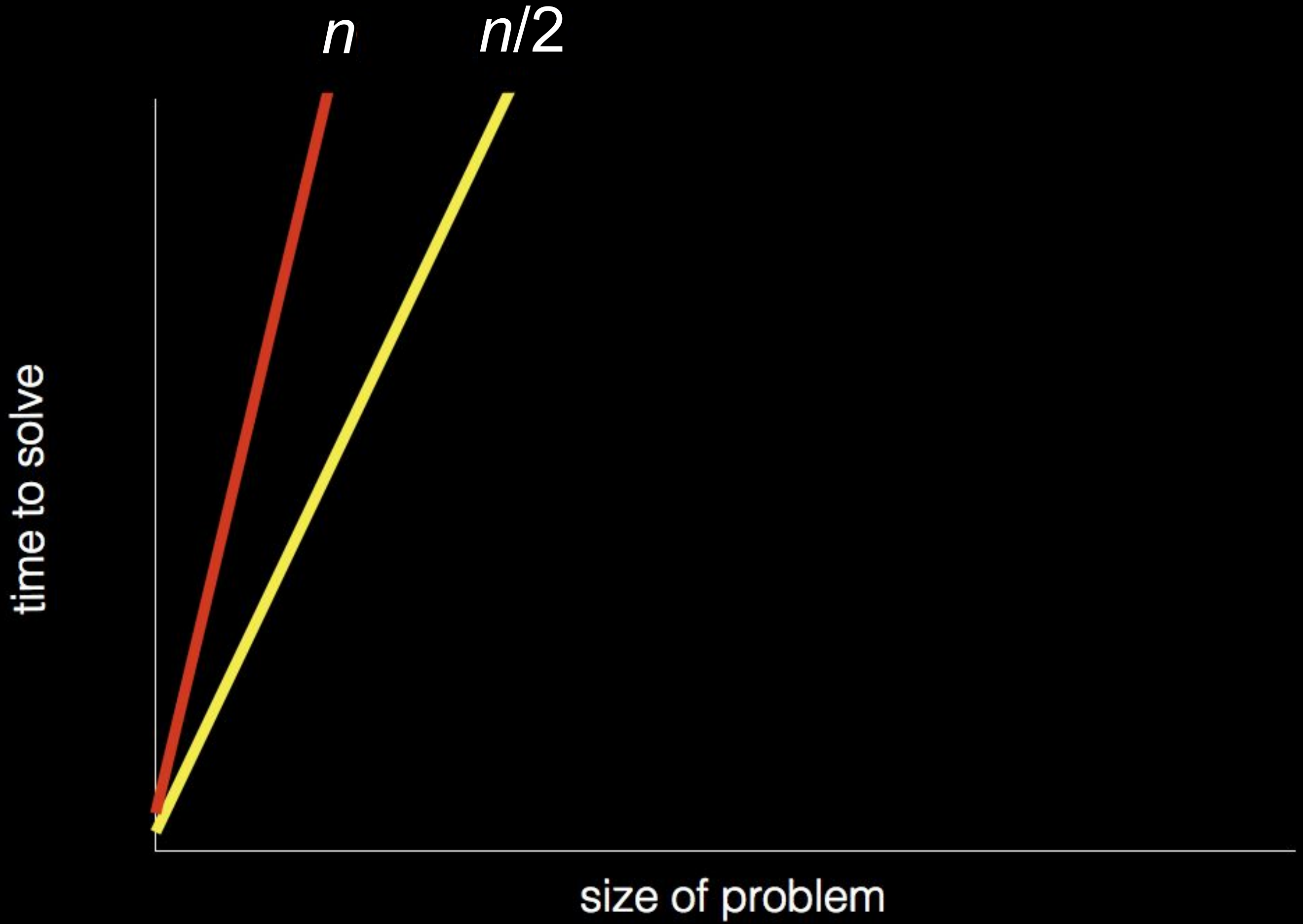
- are there any doors left to check? if not, **return!**
- check middle of list - if there, **return!**
- if search term > middle, **binary search** to the right
- if search term < middle, **binary search** to the left



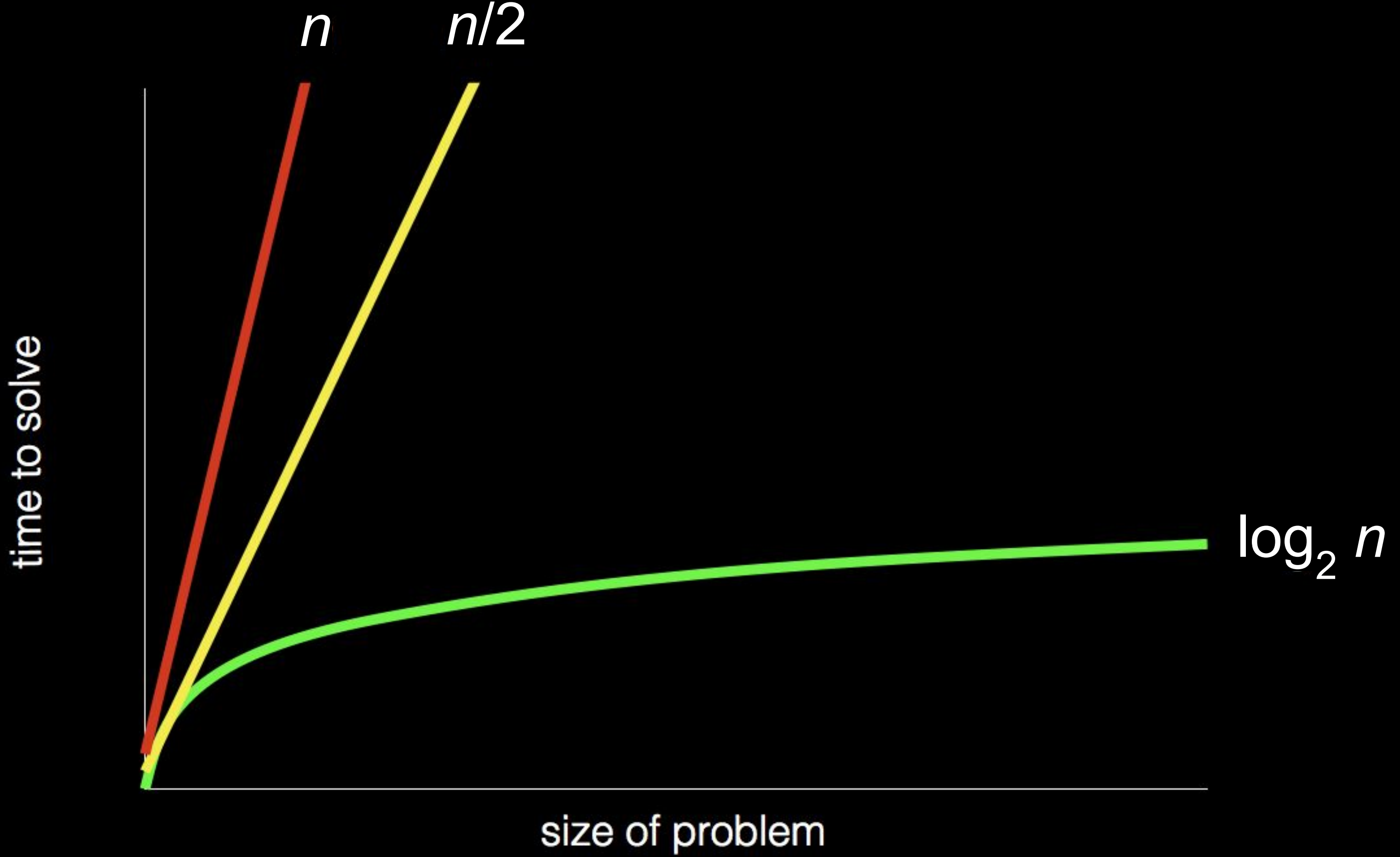
from Harvard CS50



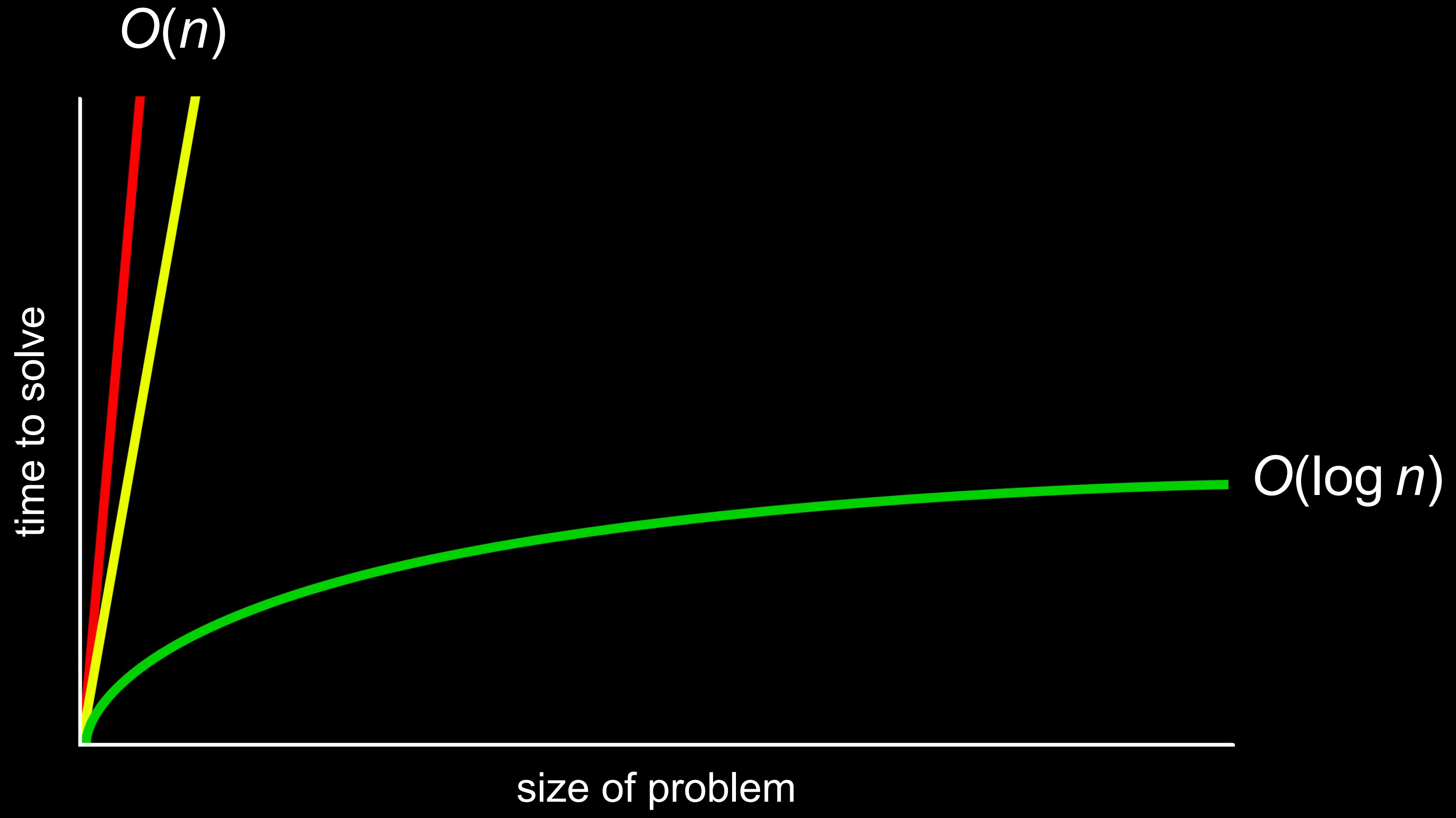
from Harvard CS50



from Harvard CS50



from Harvard CS50



from Harvard CS50

o

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

from Harvard CS50

$O(n^2)$

$O(n \log n)$

$O(n)$ linear search

$O(\log n)$

$O(1)$

from Harvard CS50

$O(n^2)$	
$O(n \log n)$	
$O(n)$	linear search
$O(\log n)$	binary search
$O(1)$	

Ω

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$ linear search

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$ linear search, binary search