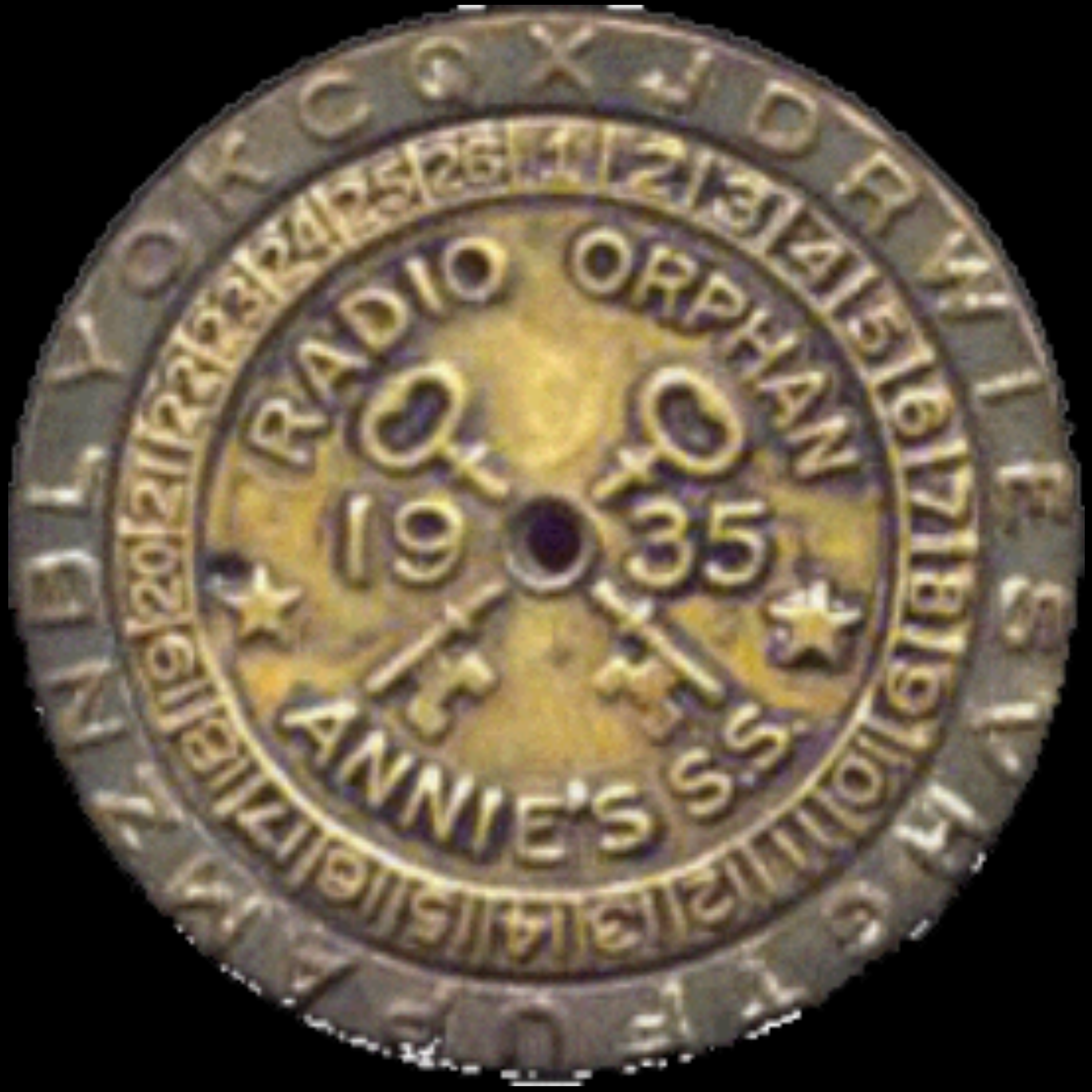
Week 3

last time

```
help50

eprintf

debug50
```



http://www.radioarchives.org/annie/

string

Z a m y l a

string

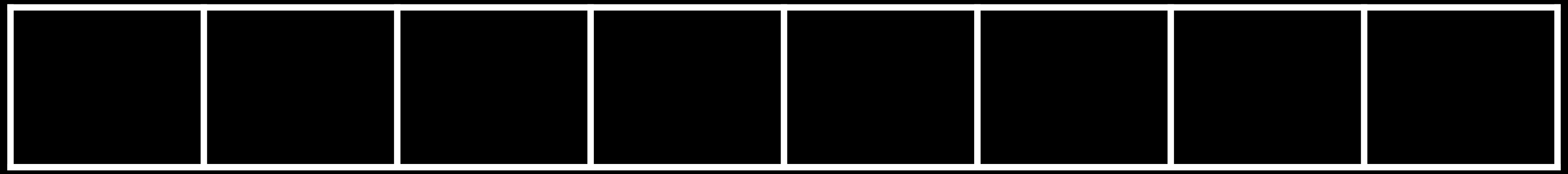
Z	a	m	y	l	a

string

Z	a	m	y	l	a	\0	

```
int main(void)
int main(int argc, string argv[])
```

this time

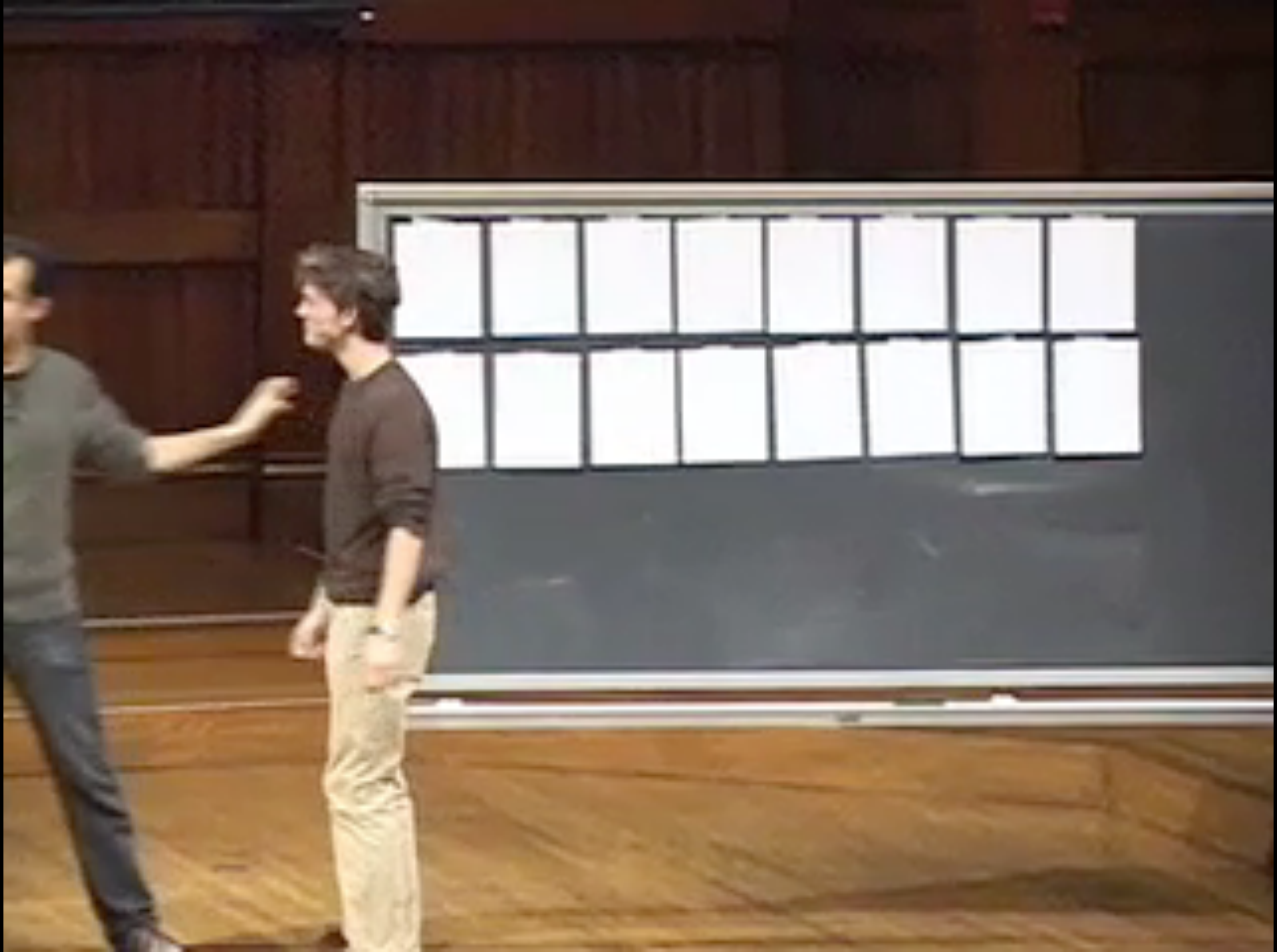


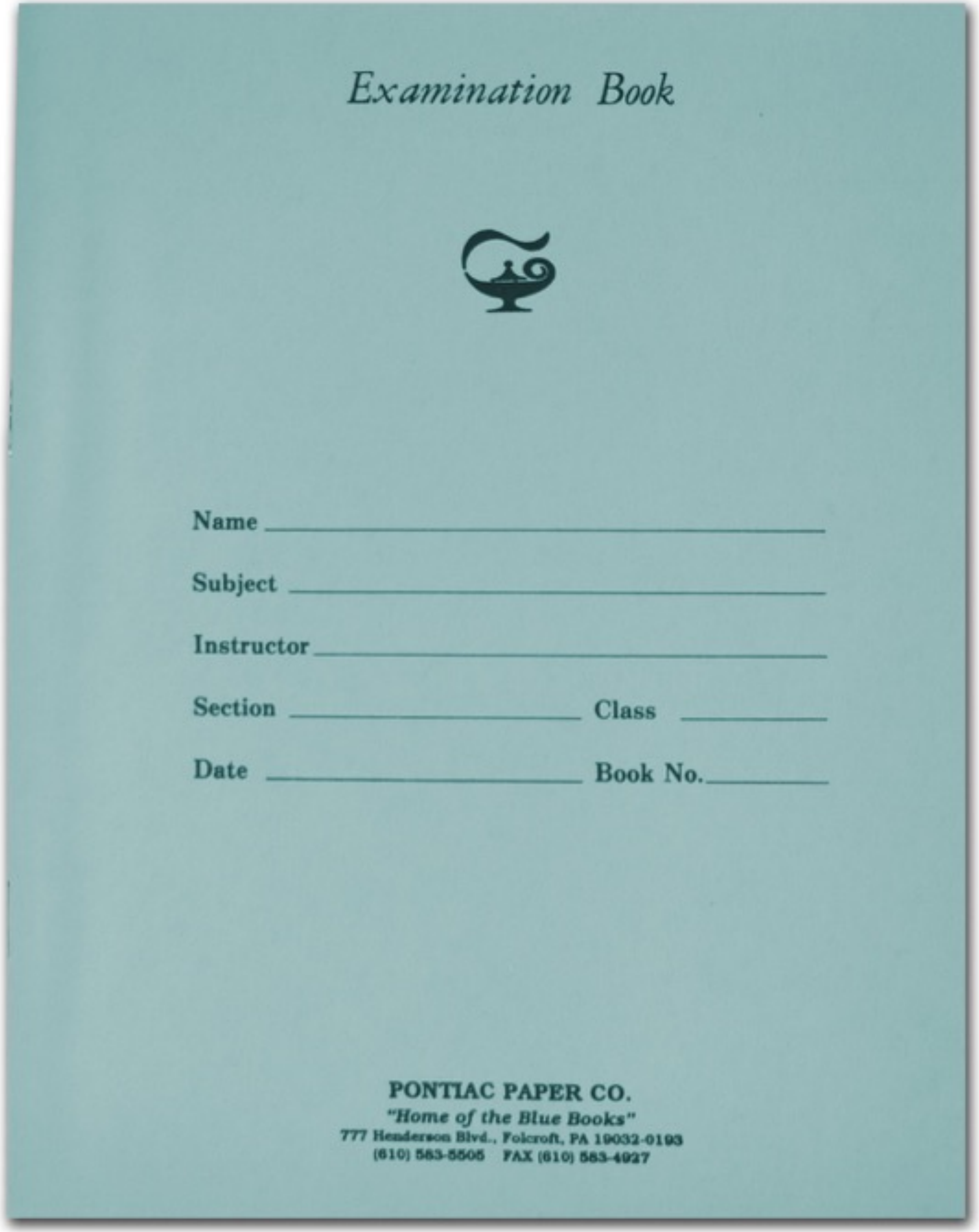
linear search

```
for each element in array
    if element you're looking for
        return true
return false
```

binary search

```
look at middle of sorted array
if element you're looking for
    return true
else if element is to left
    search left half of array
else if element is to right
    search right half of array
else
    return false
```





Examination Book

Name ______________________________

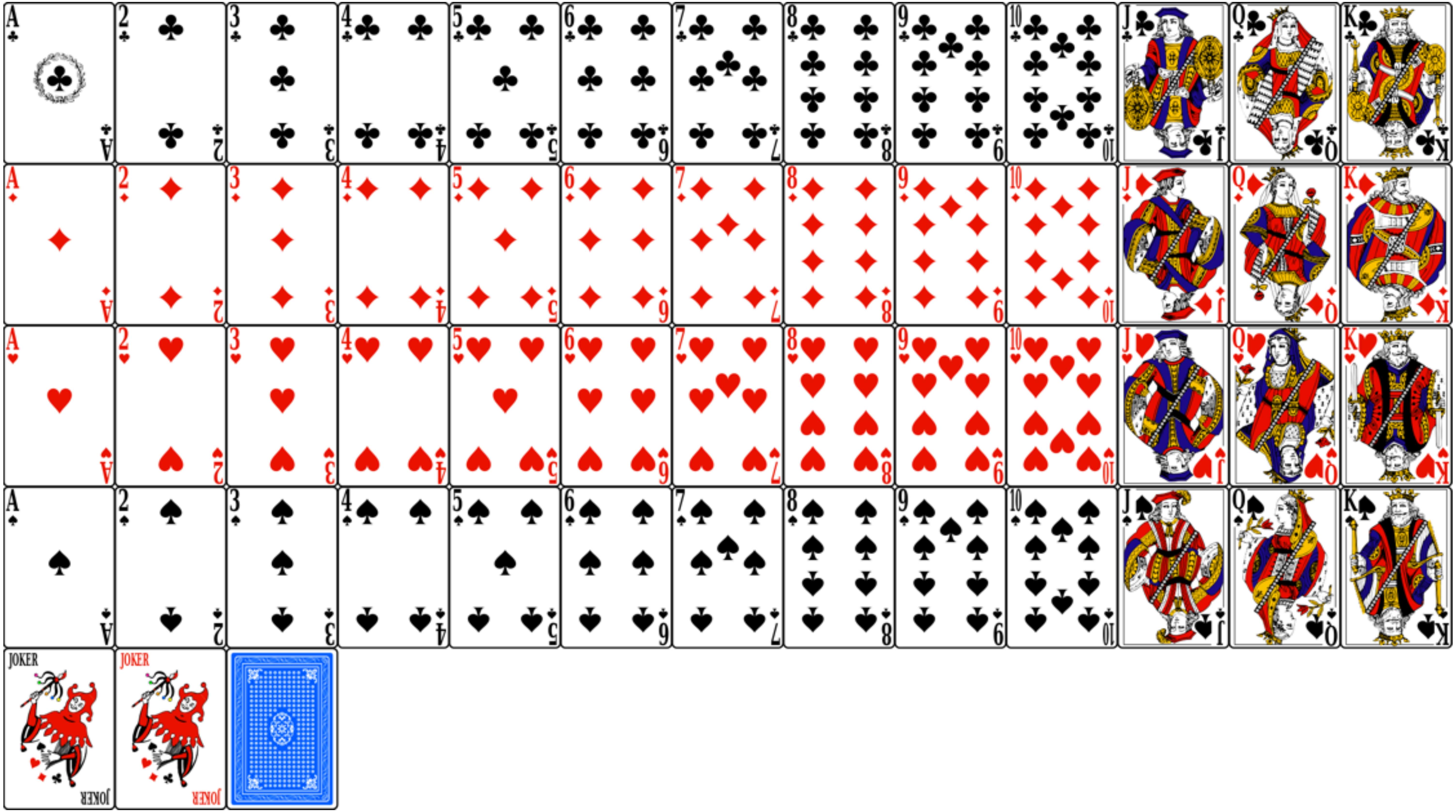
Subject ____________________________

Instructor __________________________

Section ______________ Class __________

Date ________________ Book No. ________

PONTIAC PAPER CO.
"Home of the Blue Books"
777 Henderson Blvd., Folcroft, PA 19032-0193
(610) 583-5505 FAX (610) 583-4927



wikipedia.org

4 2 6 8 1 3 7 5

bubble sort

```
repeat until no swaps
    for i from 0 to n-2
        if i'th and i+1'th elements out of order
            swap them
```

selection sort

```
for i from 0 to n-1
    find smallest element between i'th and n-1'th
    swap smallest with i'th element
```

insertion sort

```
for i from 1 to n-1
    call 0'th through i-1'th elements the "sorted side"
    remove i'th element
    insert it into sorted side in order
```

running time

bubble sort

$(n - 1)$

$$(n - 1) + (n - 2)$$

$$(n - 1) + (n - 2) + \ldots + 1$$

$$(n - 1) + (n - 2) + \ldots + 1$$

$$n(n - 1)/2$$

$$(n - 1) + (n - 2) + \ldots + 1$$

$$n(n - 1)/2$$

$$(n^2 - n)/2$$

$$(n - 1) + (n - 2) + \ldots + 1$$

$$n(n - 1)/2$$

$$(n^2 - n)/2$$

$$n^2/2 - n/2$$

1,000,000

$$n^2/2 - n/2$$

$$n^2/2 - n/2$$

$$1{,}000{,}000^2/2 - 1{,}000{,}000/2$$

$$n^2/2 - n/2$$

$$1{,}000{,}000^2/2 - 1{,}000{,}000/2$$

$$500{,}000{,}000{,}000 - 500{,}000$$

$$n^2/2 - n/2$$

$$1{,}000{,}000^2/2 - 1{,}000{,}000/2$$

$$500{,}000{,}000{,}000{,}000 - 500{,}000$$

$$499{,}999{,}500{,}000$$

$$n^2/2 - n/2$$

$$O(n^2)$$

O

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

…

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

…

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

…

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

…

$O(n^2)$

$O(n \log n)$

$O(n)$

$O(\log n)$

$O(1)$

…

Ω

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

…

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

…

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

…

$\Omega(n^2)$

$\Omega(n \log n)$

$\Omega(n)$

$\Omega(\log n)$

$\Omega(1)$

…

Θ

$$O(n^2)$$

```
 0  pick up phone book
 1  open to middle of phone book
 2  look at names
 3  if Smith is among names
 4      call Mike
 5  else if Smith is earlier in book
 6      open to middle of left half of book
 7      go back to step 2
 8  else if Smith is later in book
 9      open to middle of right half of book
10      go back to step 2
11  else
12      quit
```

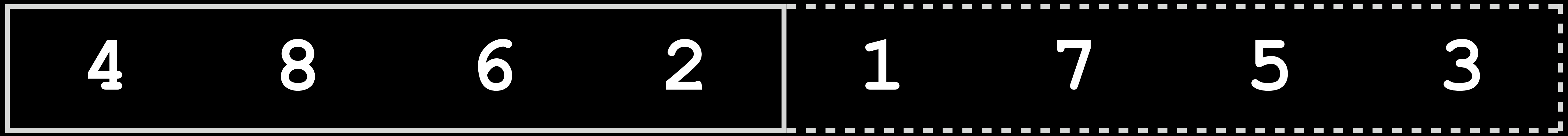
```
pick up phone book
open to middle of phone book
look at names
if Smith is among names
    call Mike
else if Smith is earlier in book
    search for Mike in left half of book

else if Smith is later in book
    search for Mike in right half of book

else
    quit
```

merge sort

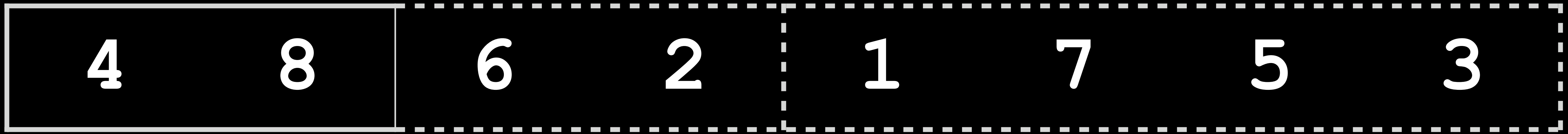
```
On input of n elements
    if $n$ < 2
        return
    else
        sort left half of elements
        sort right half of elements
        merge sorted halves
```

4 8 6 2 1 7 5 3

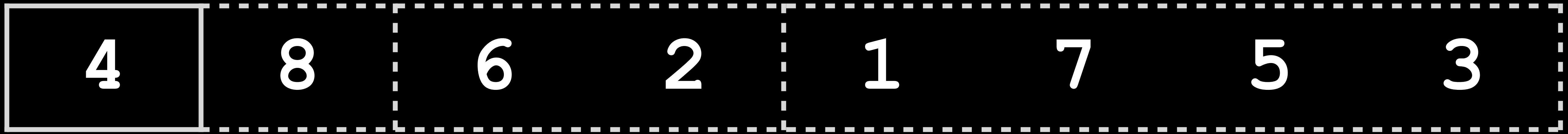
4 8 6 2 1 7 5 3



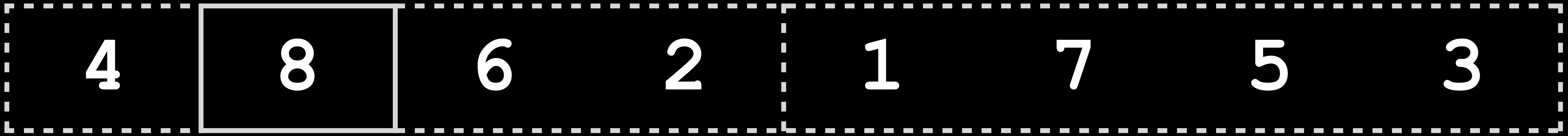
4 8 6 2 1 7 5 3



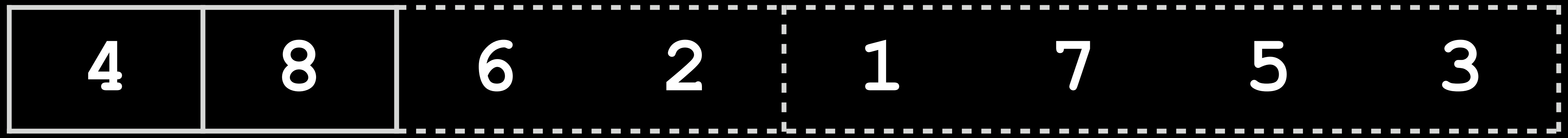
4 8 6 2 1 7 5 3



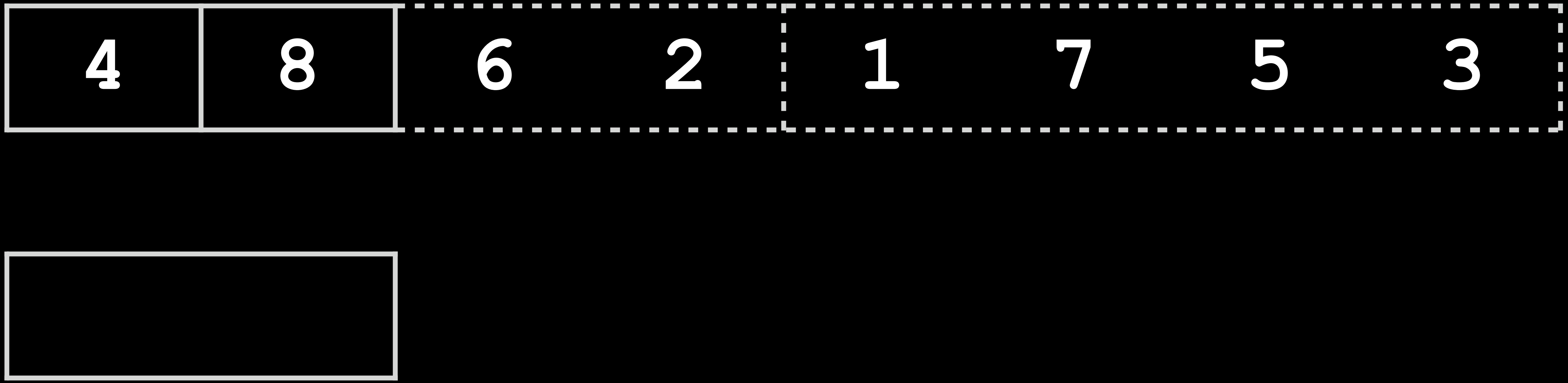
4 8 6 2 1 7 5 3



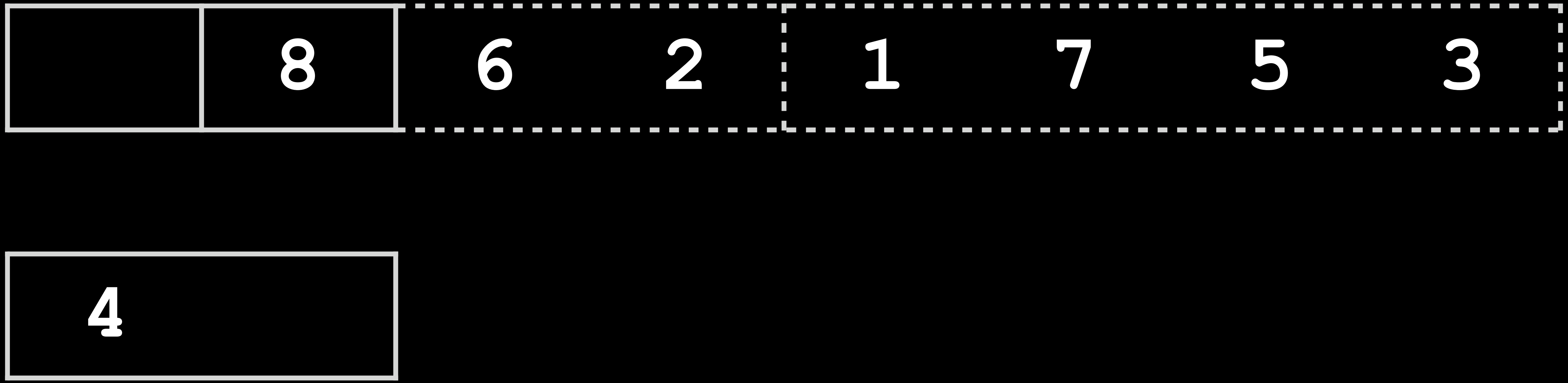
4	8	6	2	1	7	5	3



4 8 6 2 1 7 5 3

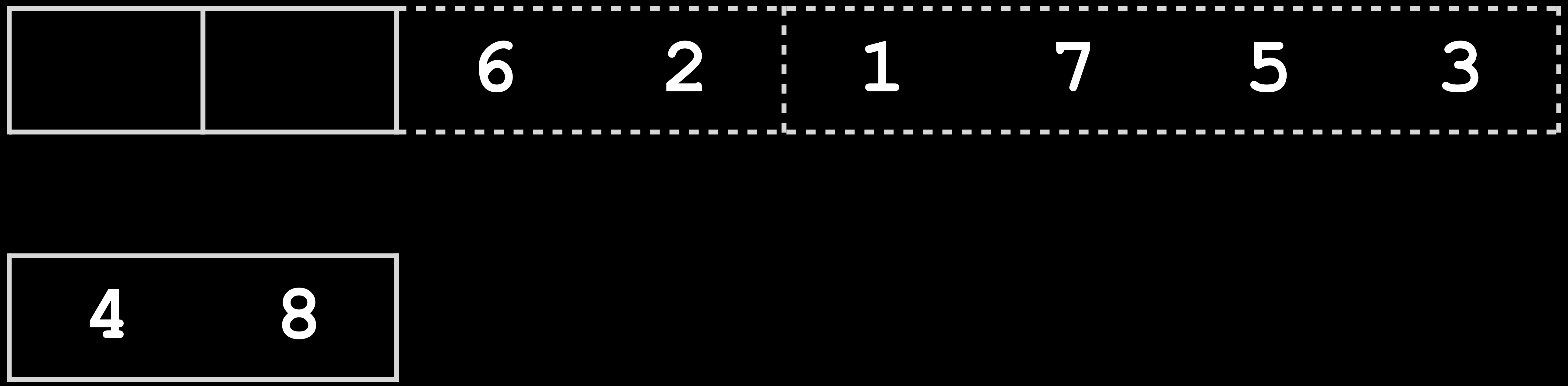


4	8	6	2	1	7	5	3



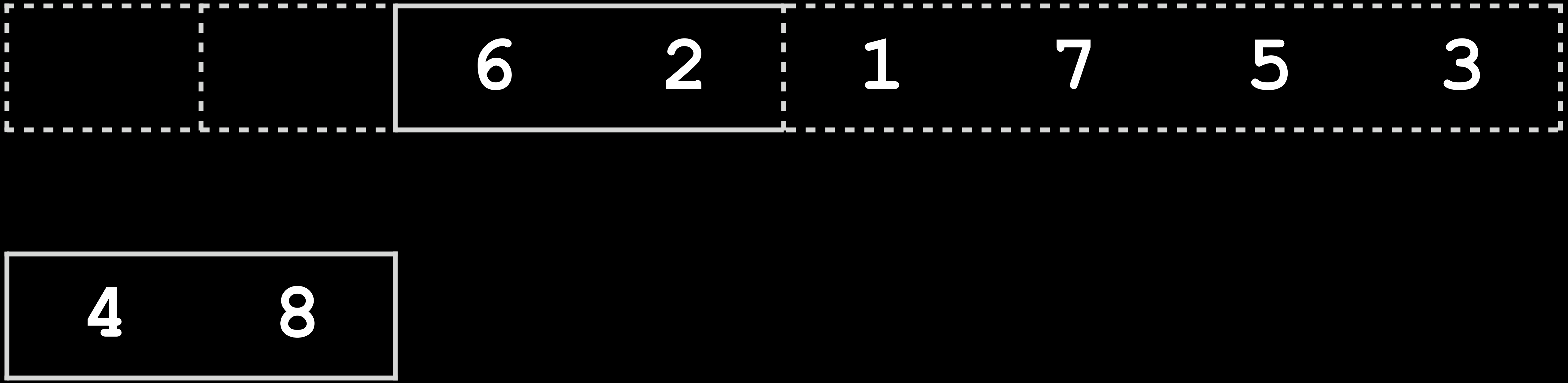
8 6 2 1 7 5 3

4



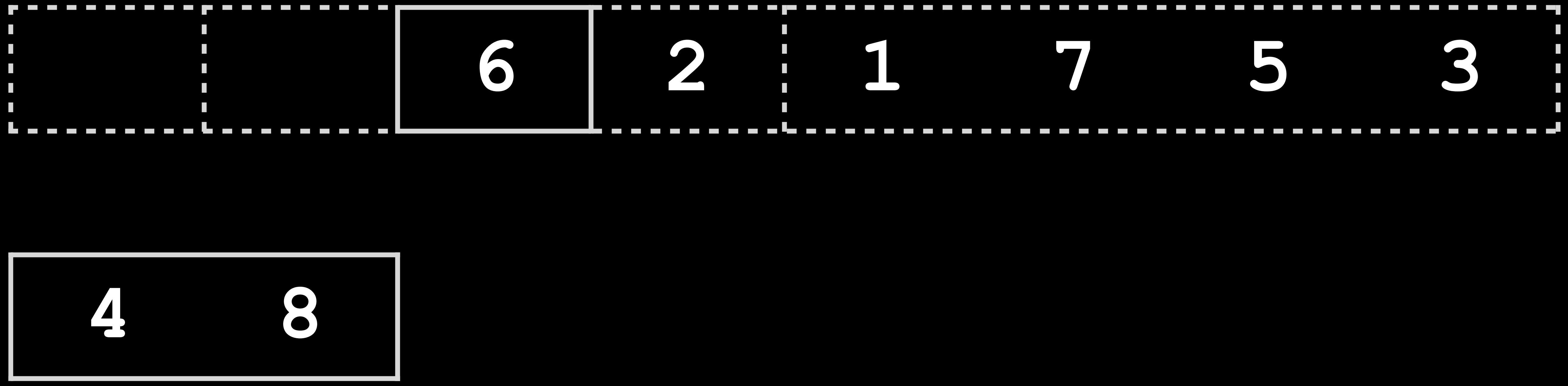
6 2 1 7 5 3

4 8



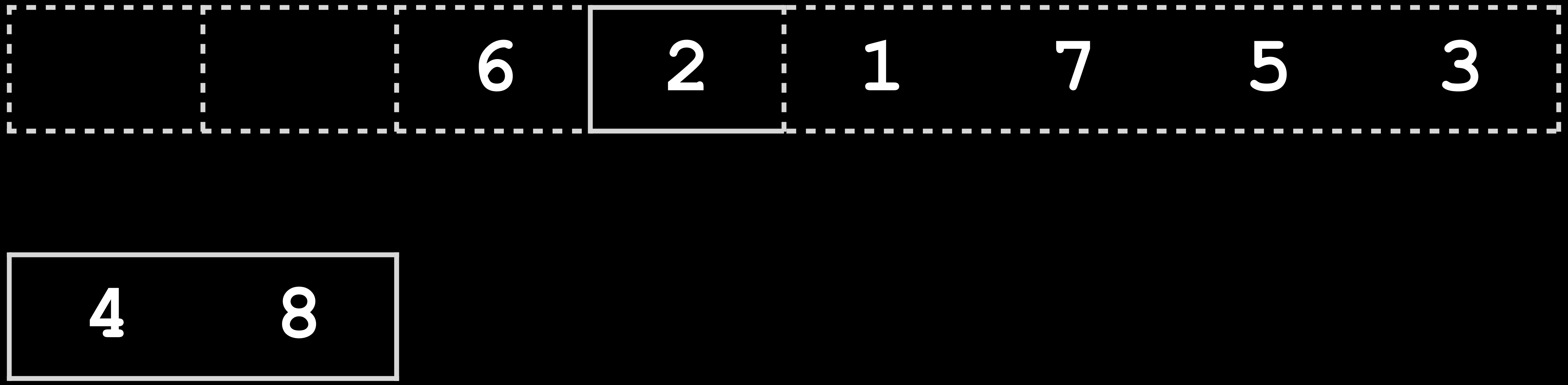
6 2 1 7 5 3

4 8



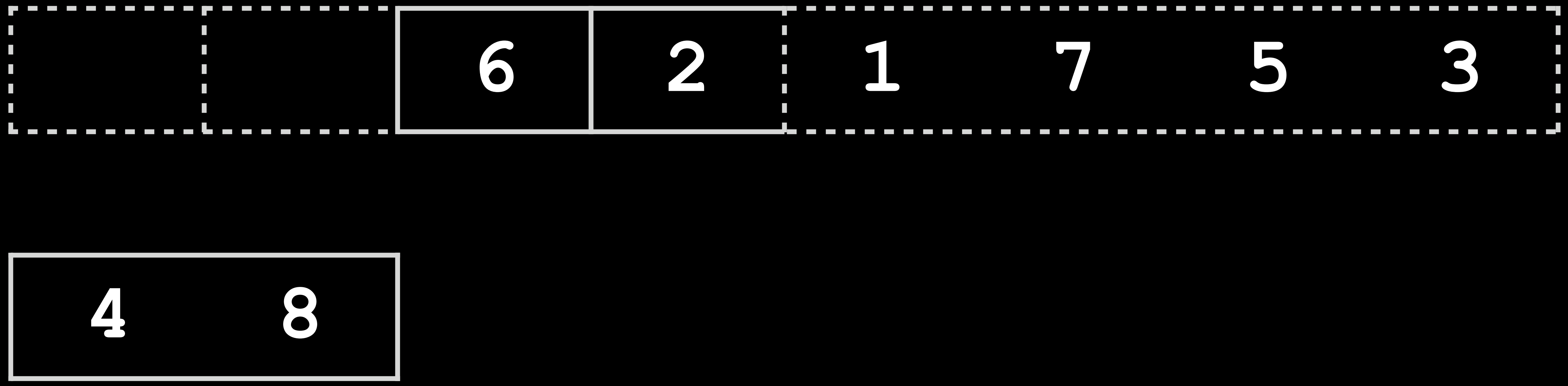
6 2 1 7 5 3

4 8



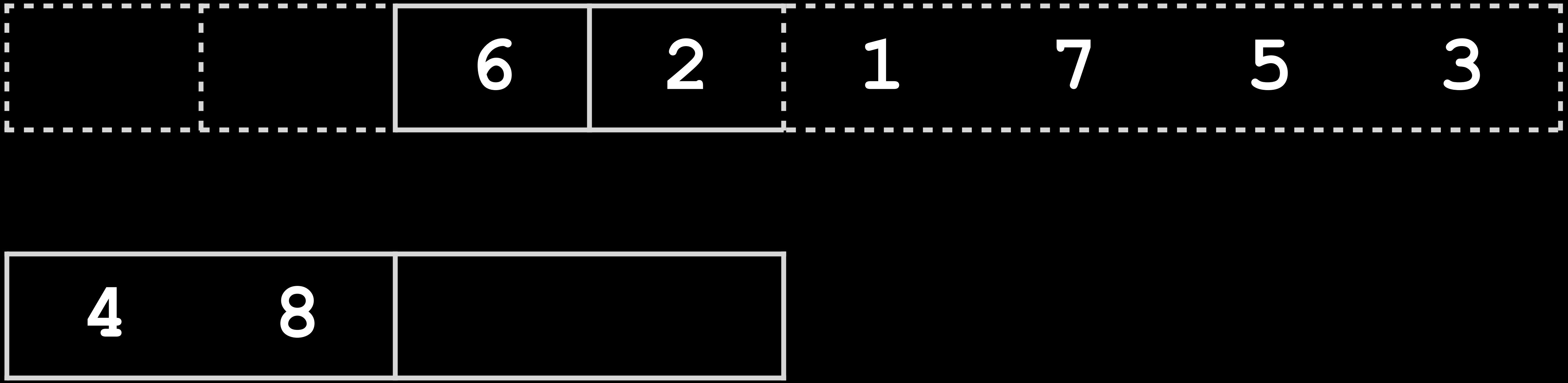
		6	2	1	7	5	3

4	8



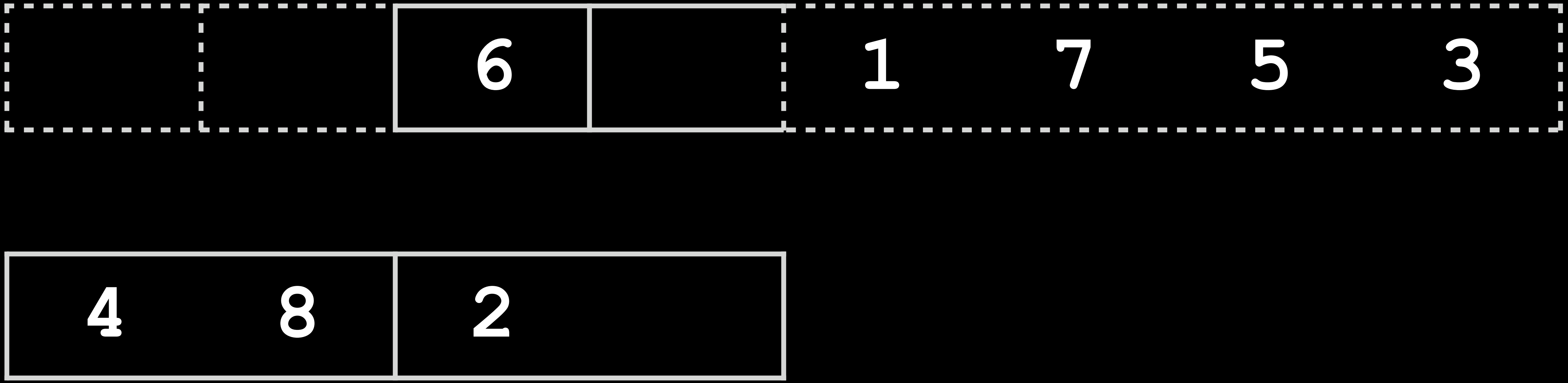
6 2 1 7 5 3

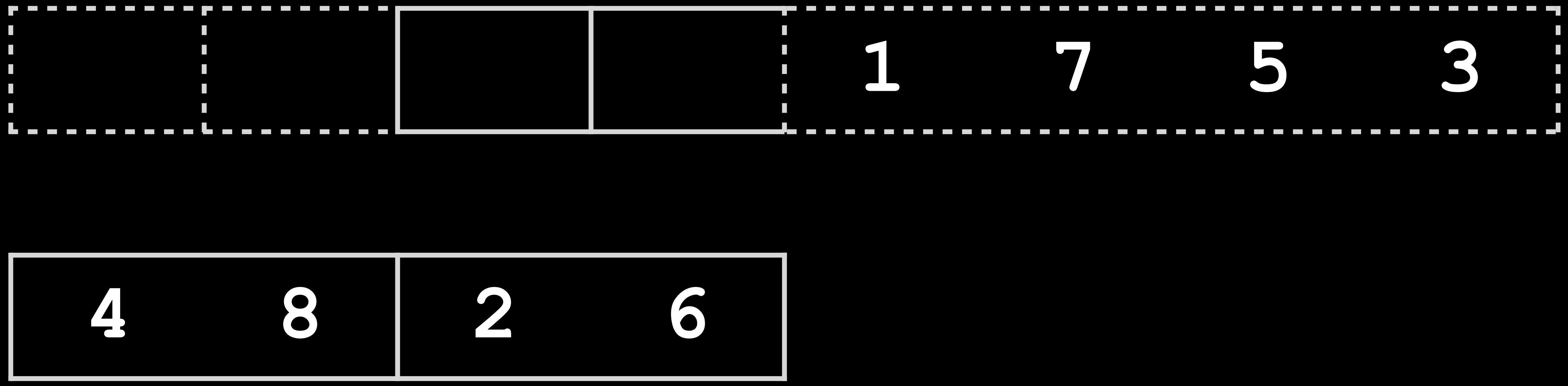
4 8



6 2 1 7 5 3

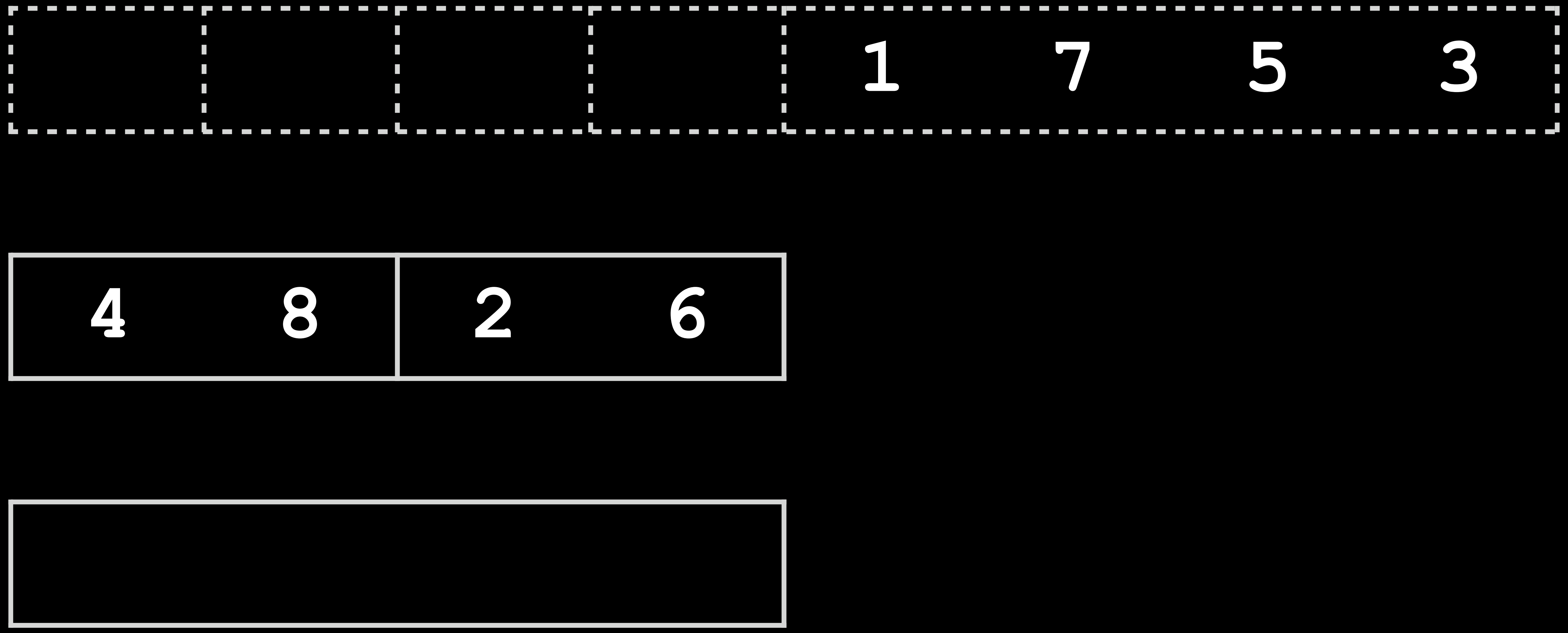
4 8

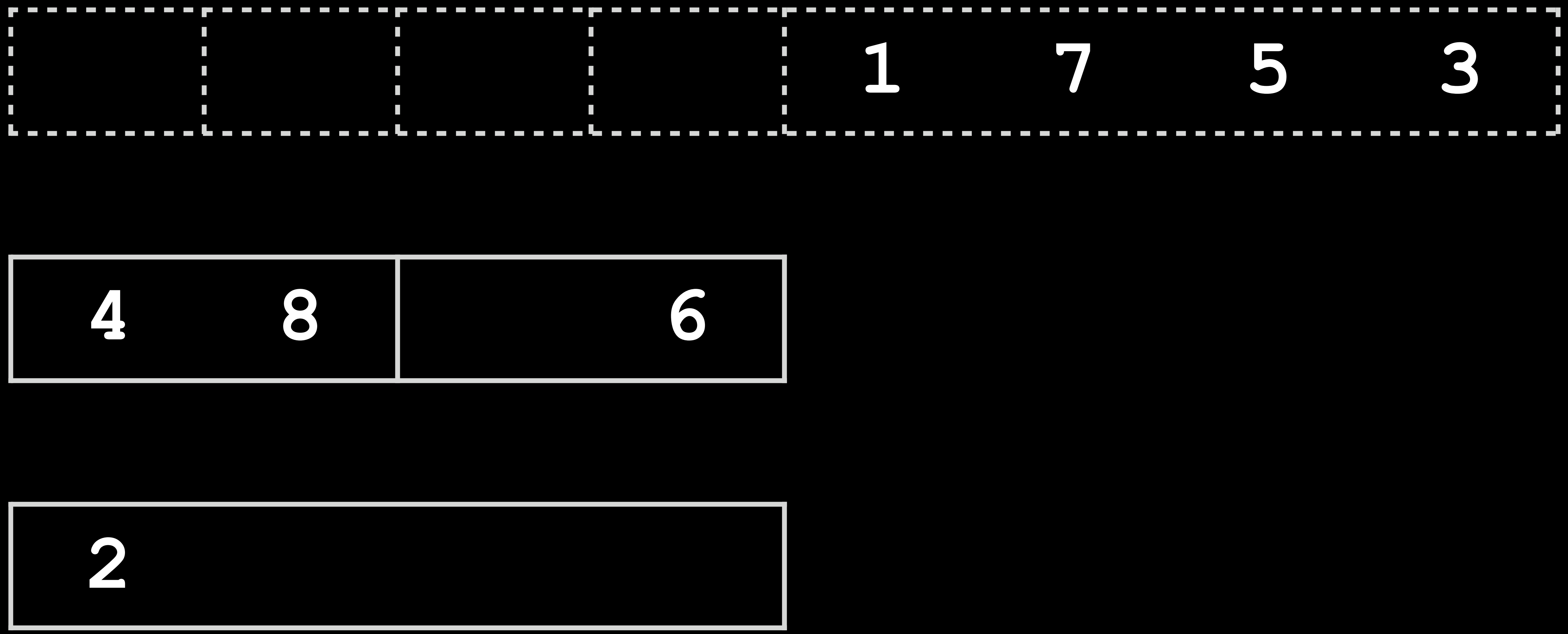

6
1
7
5
3
4
8
2



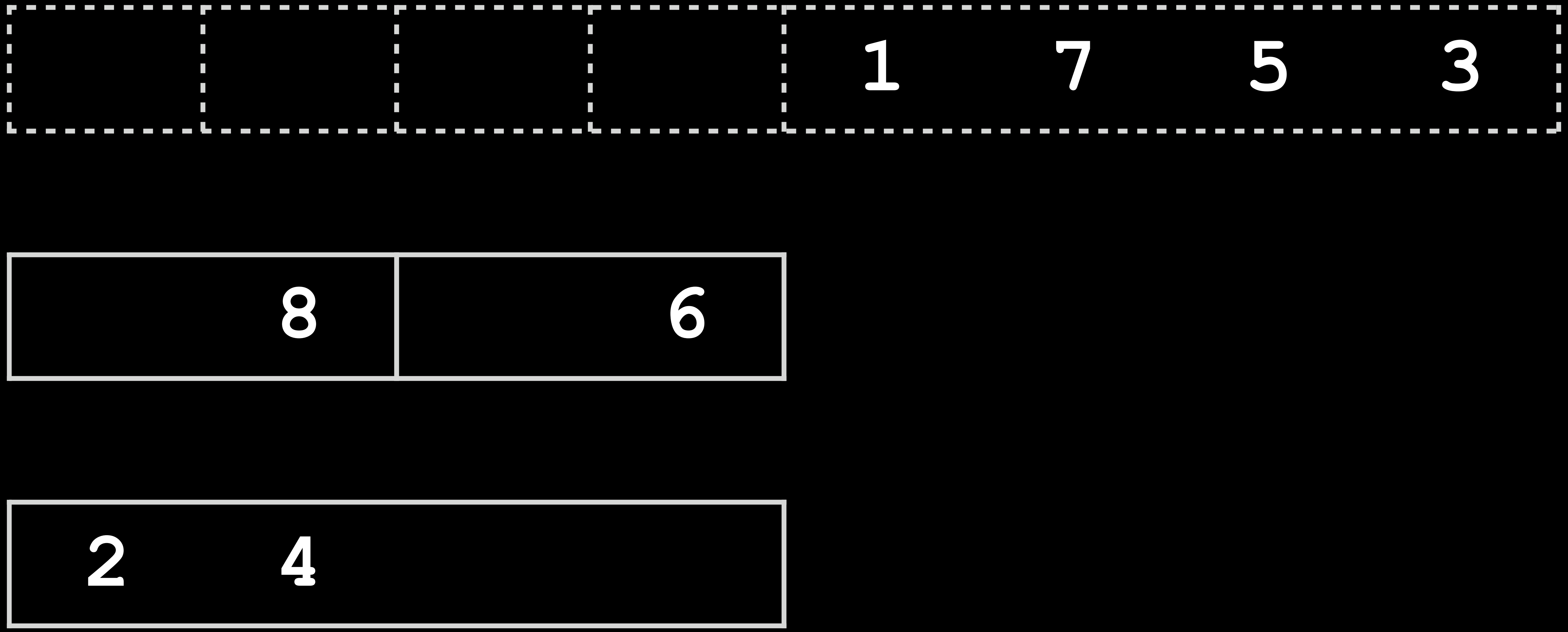
				1	7	5	3

4	8	2	6


1 7 5 3
4 8
2 6



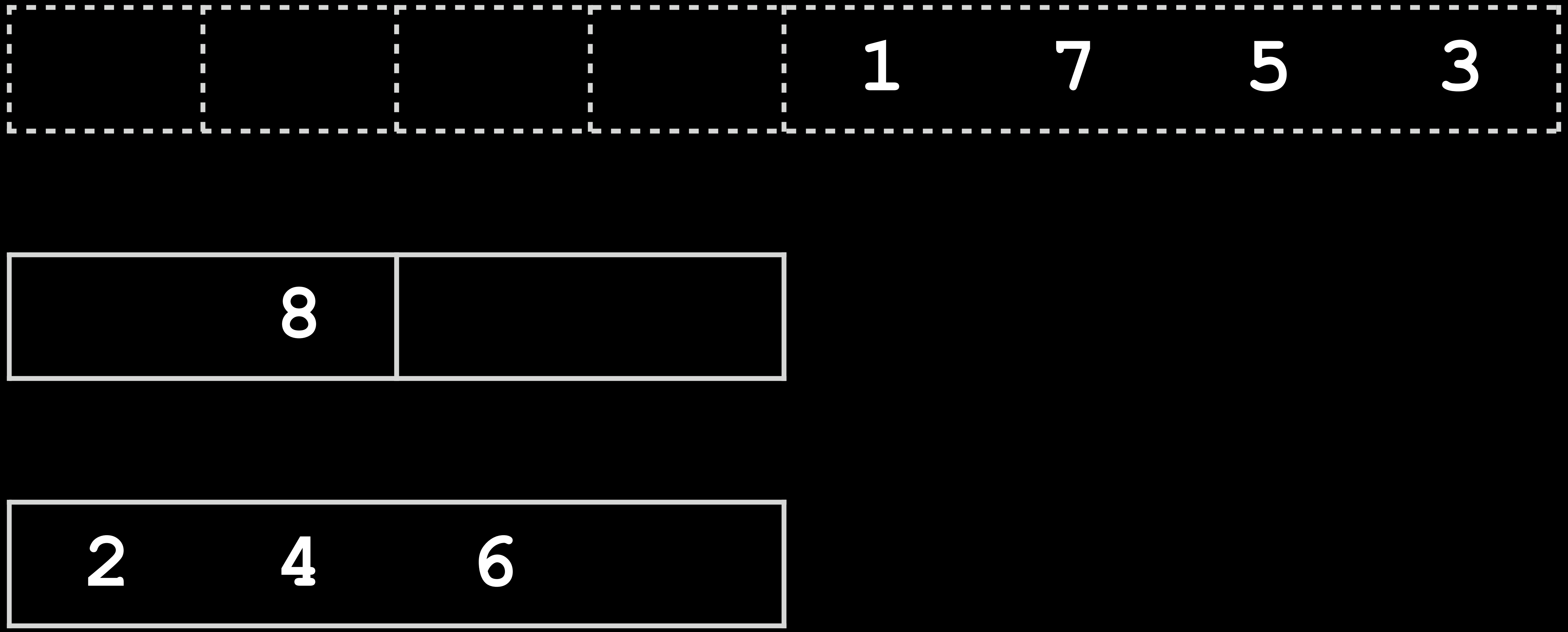
1 7 5 3
4 8
6
2

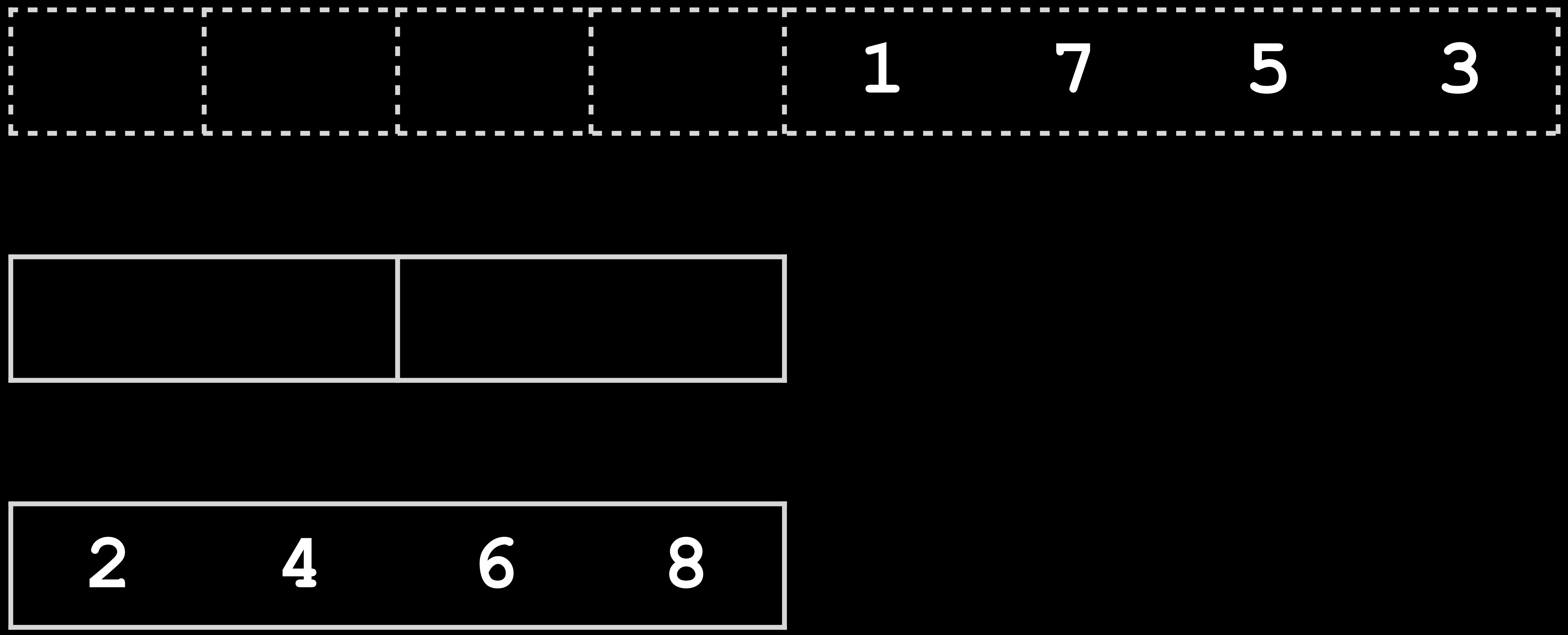


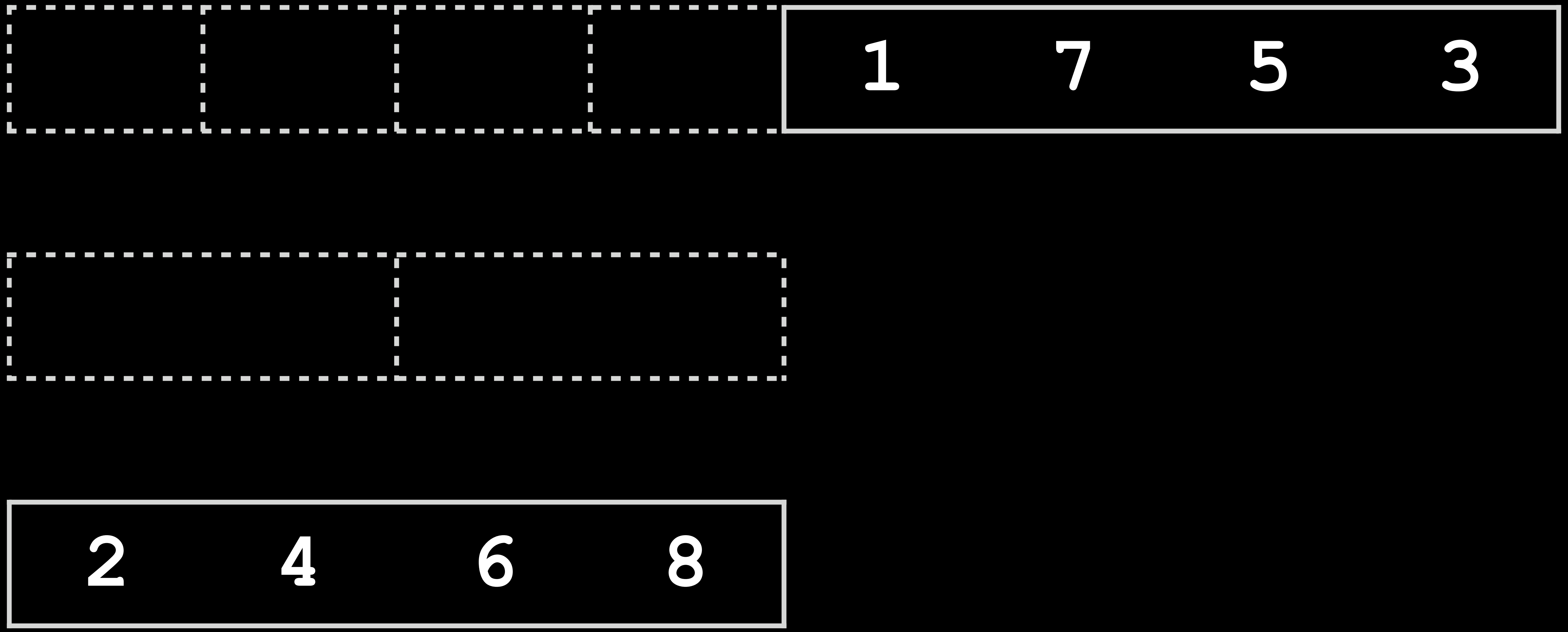
1 7 5 3

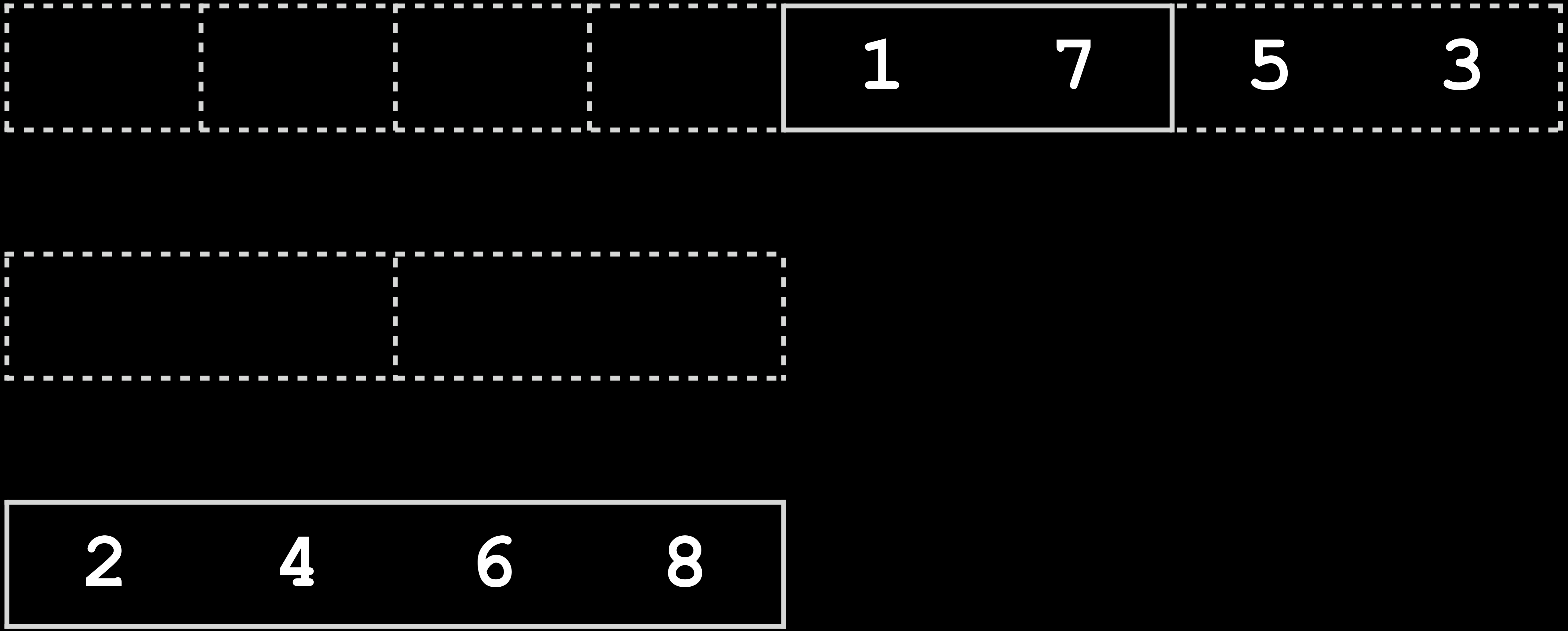
8 6

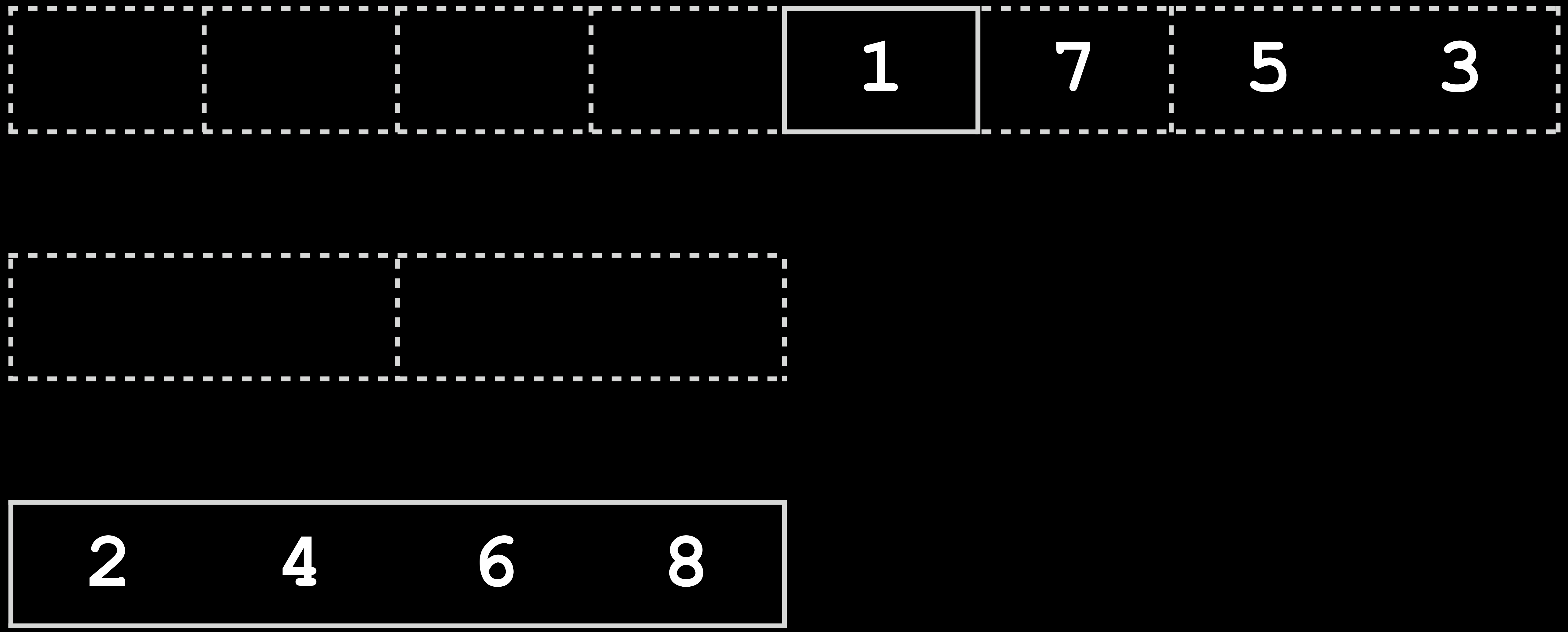
2 4

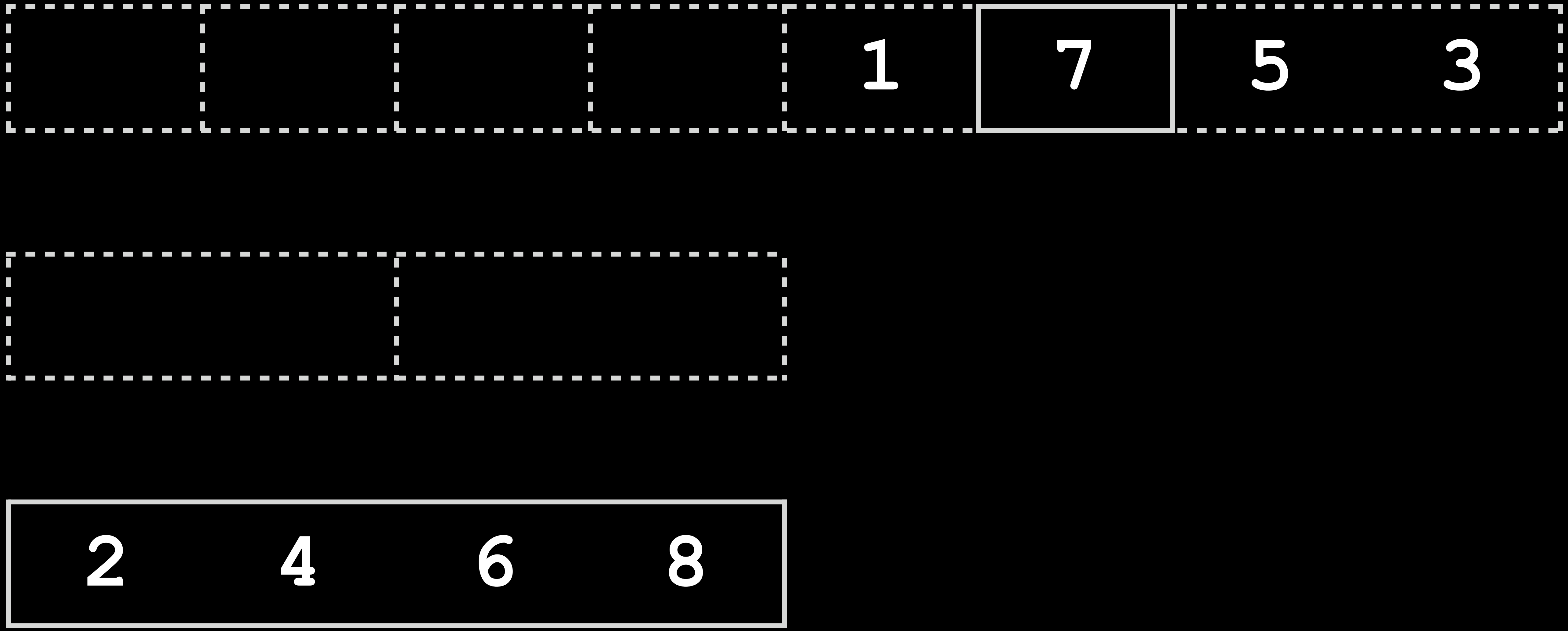

1
7
5
3
8
2
4
6

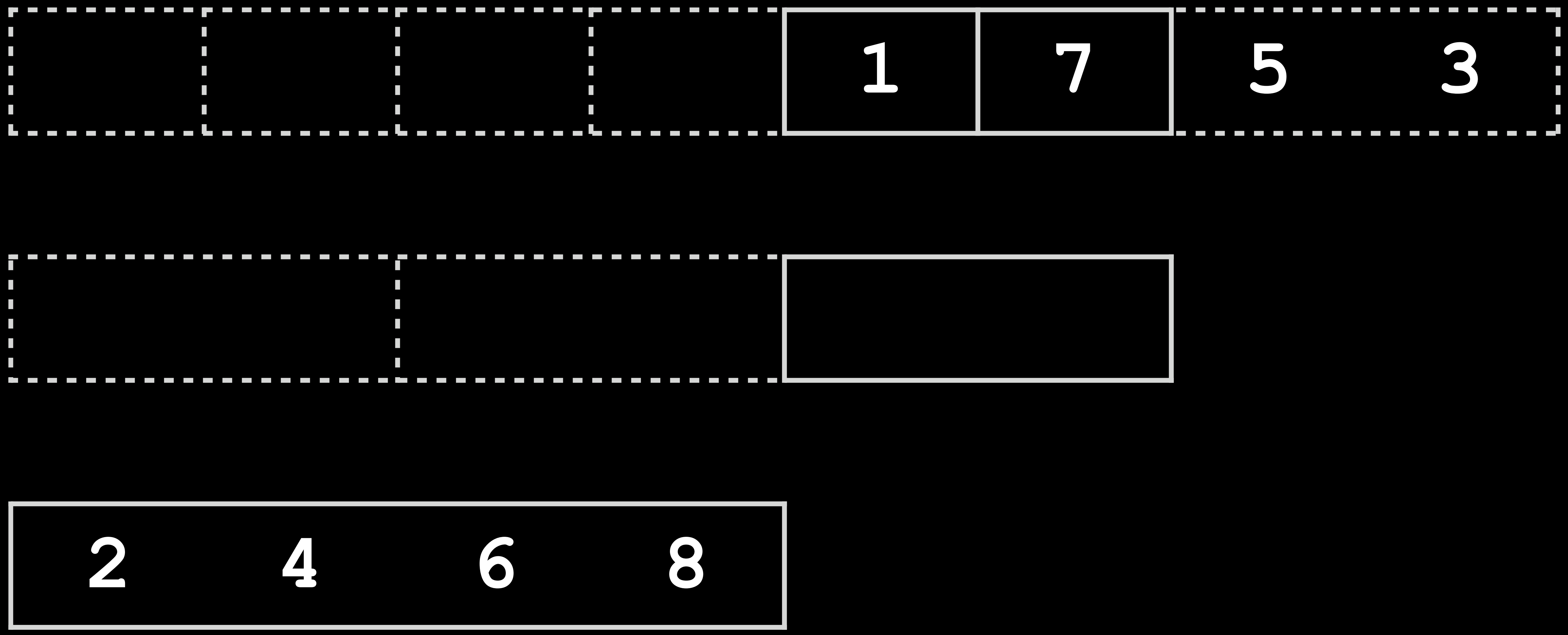

1 7 5 3
2 4 6 8


1 7 5 3
2 4 6 8


1 7
5 3
2 4 6 8

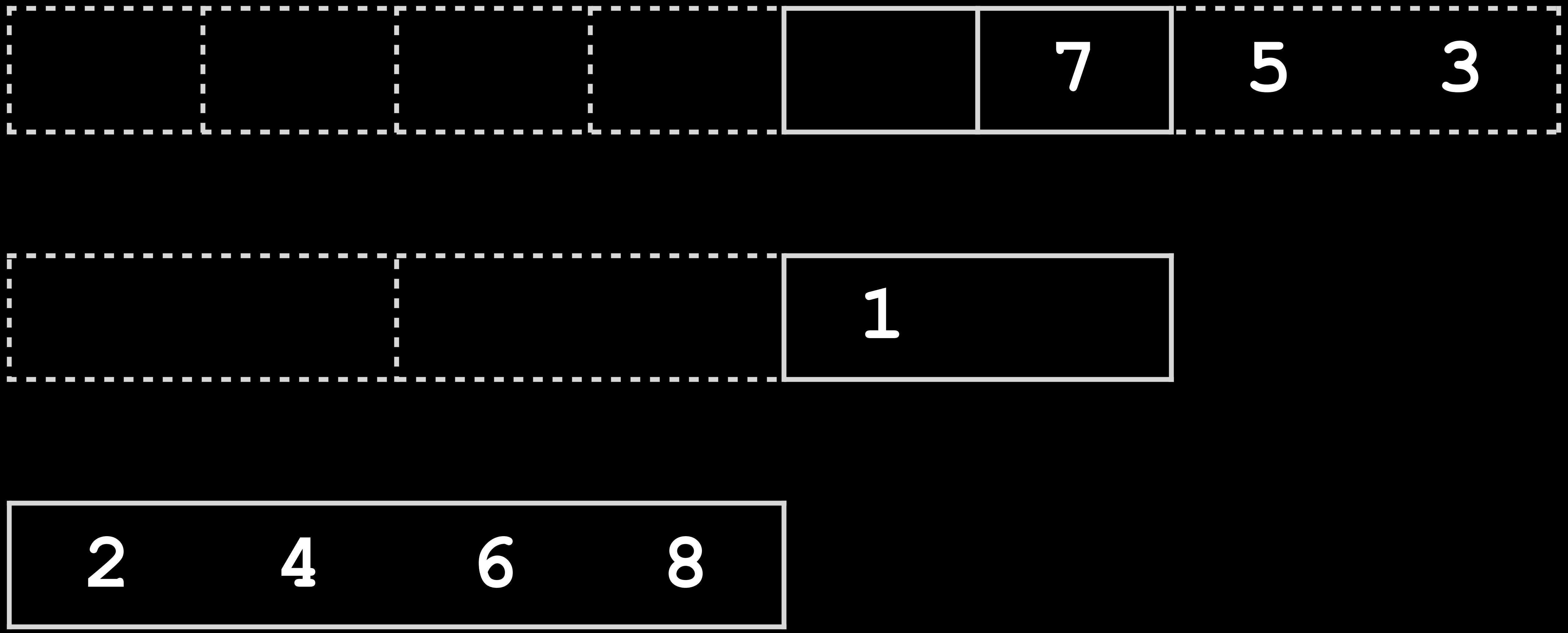

1
7
5
3
2
4
6
8


1
7
5
3
2
4
6
8

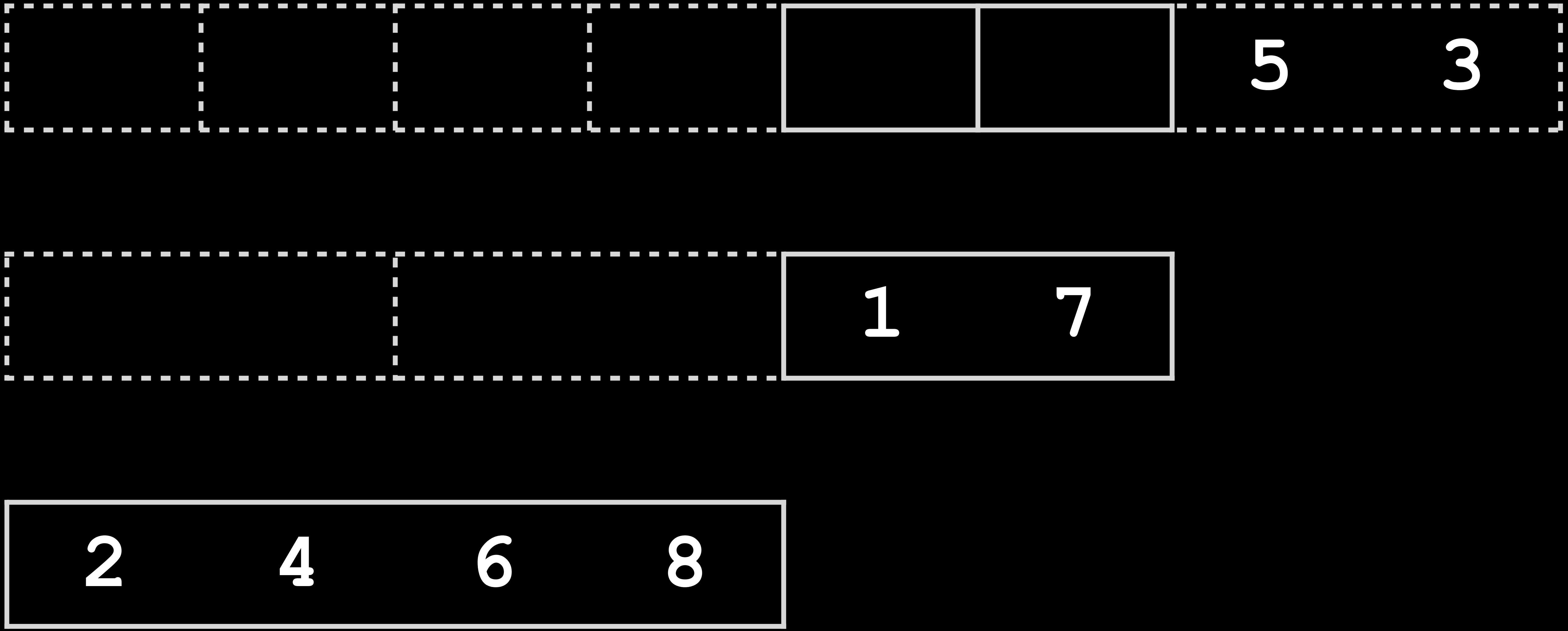


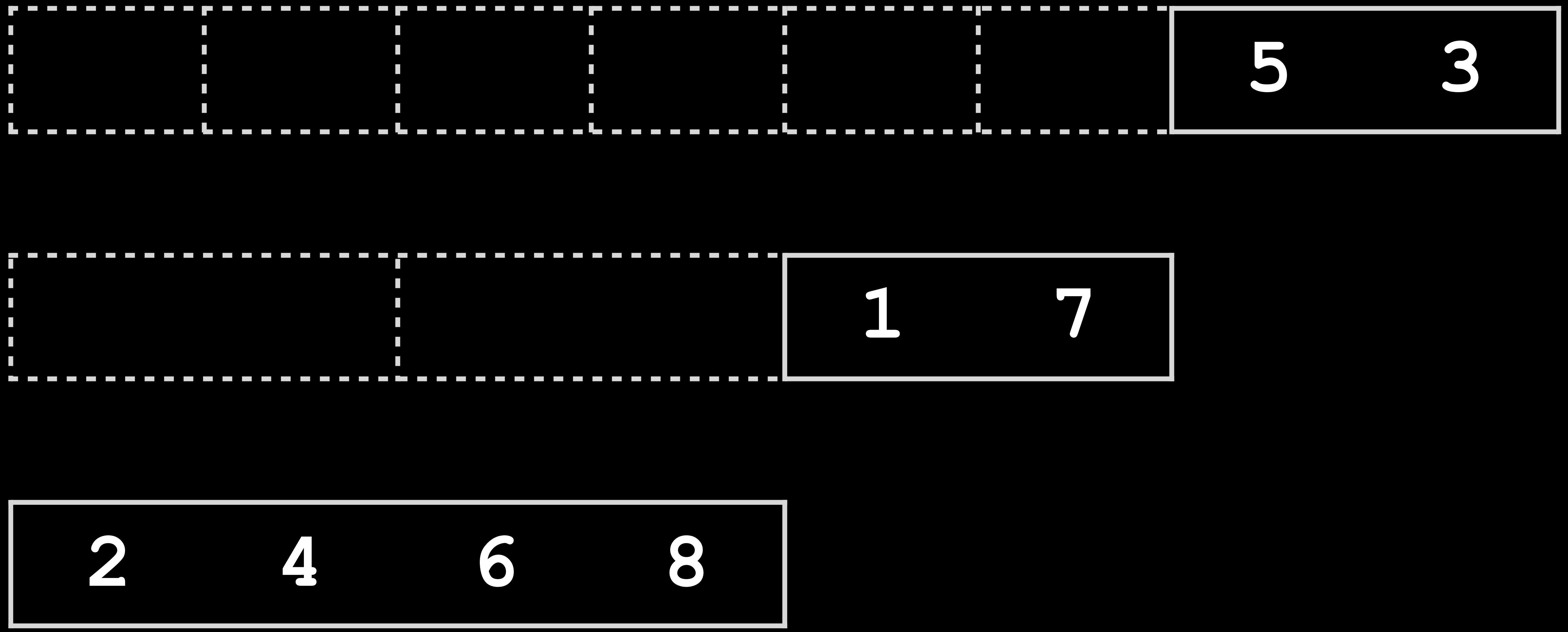
1 7 5 3

2 4 6 8



7 5 3
1
2 4 6 8

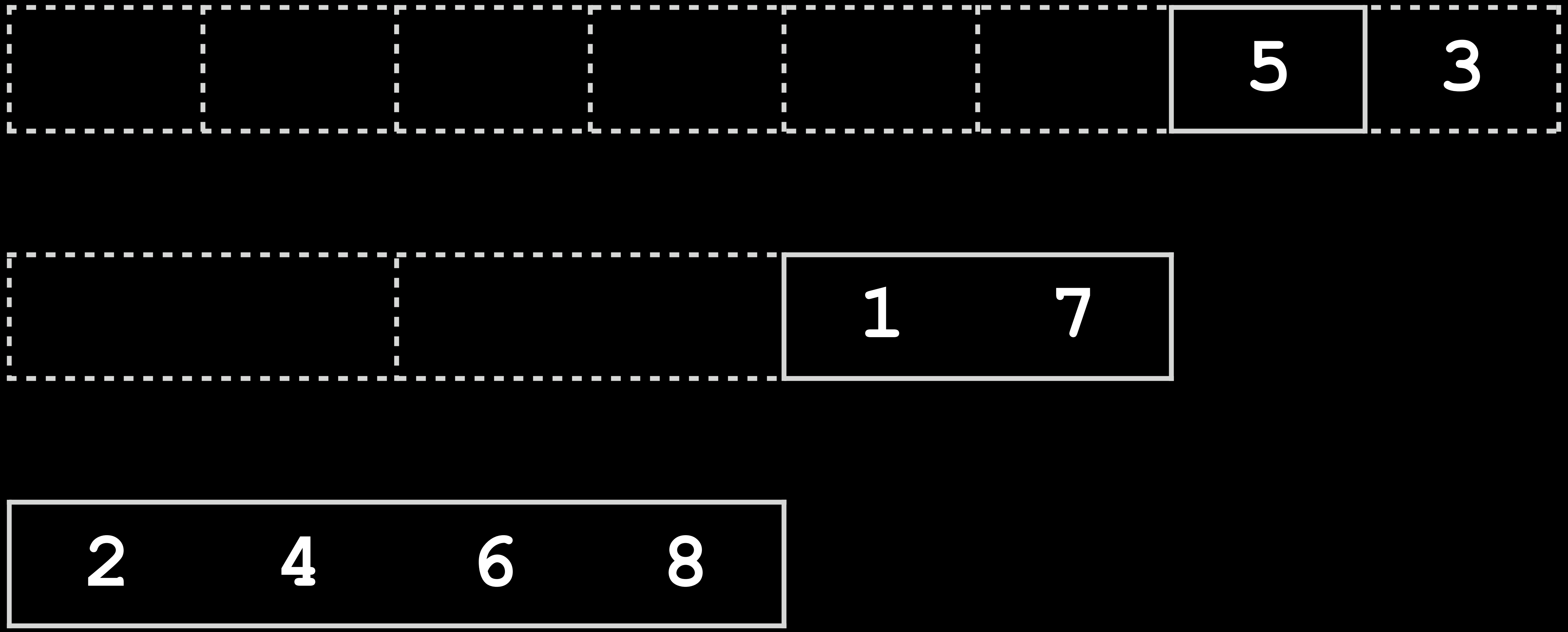

5 3
1 7
2 4 6 8



5 3

1 7

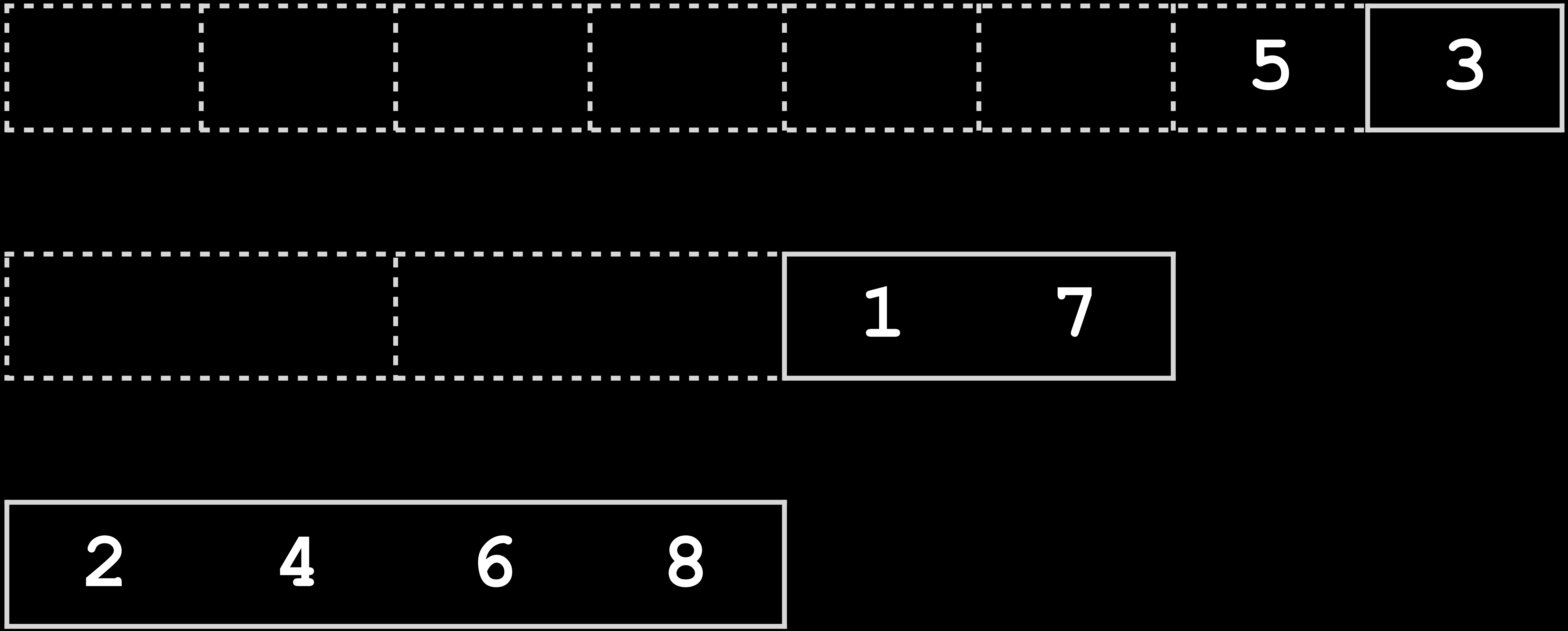
2 4 6 8



5 3

1 7

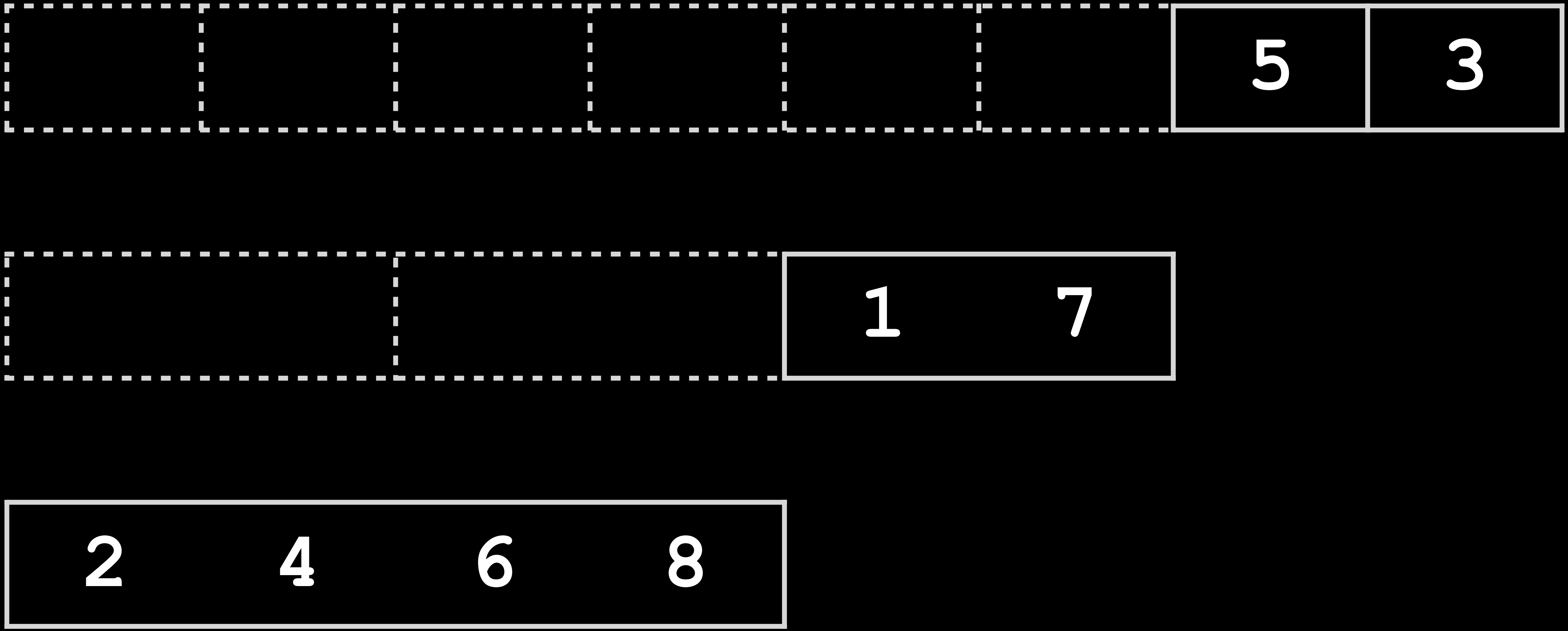
2 4 6 8

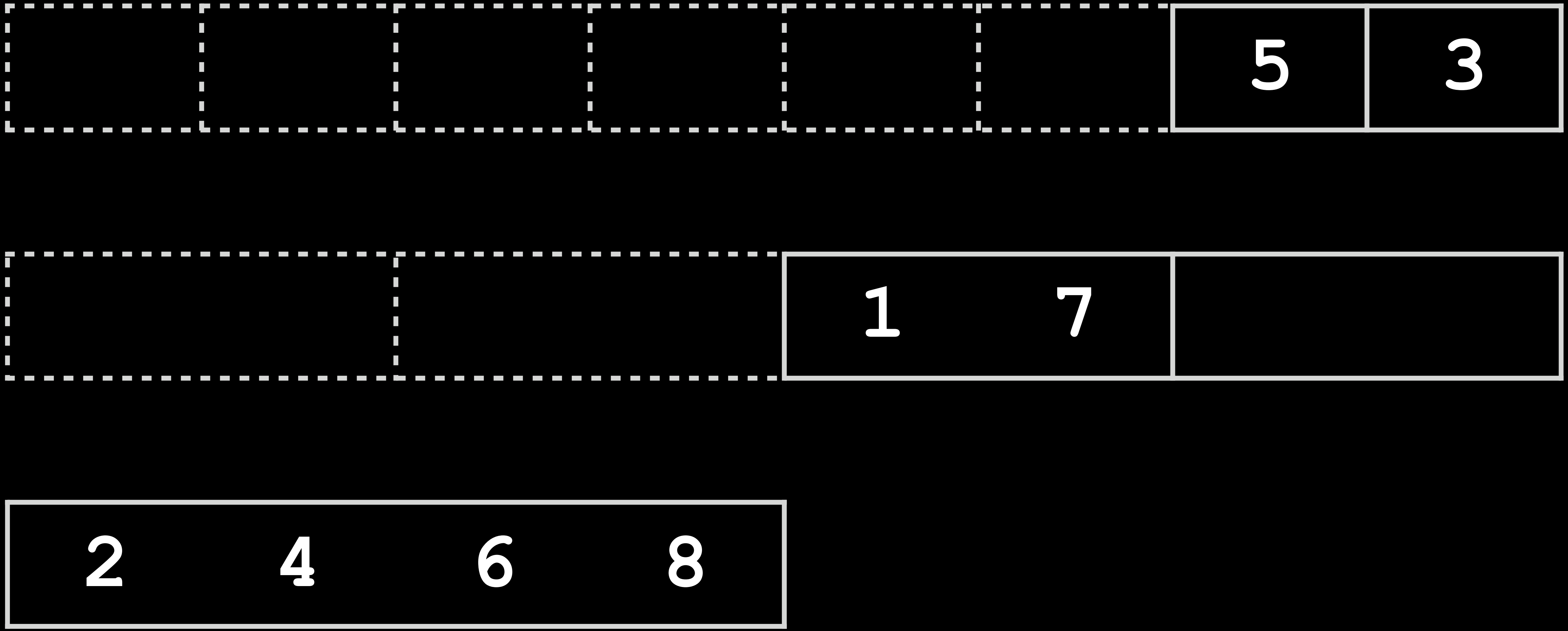


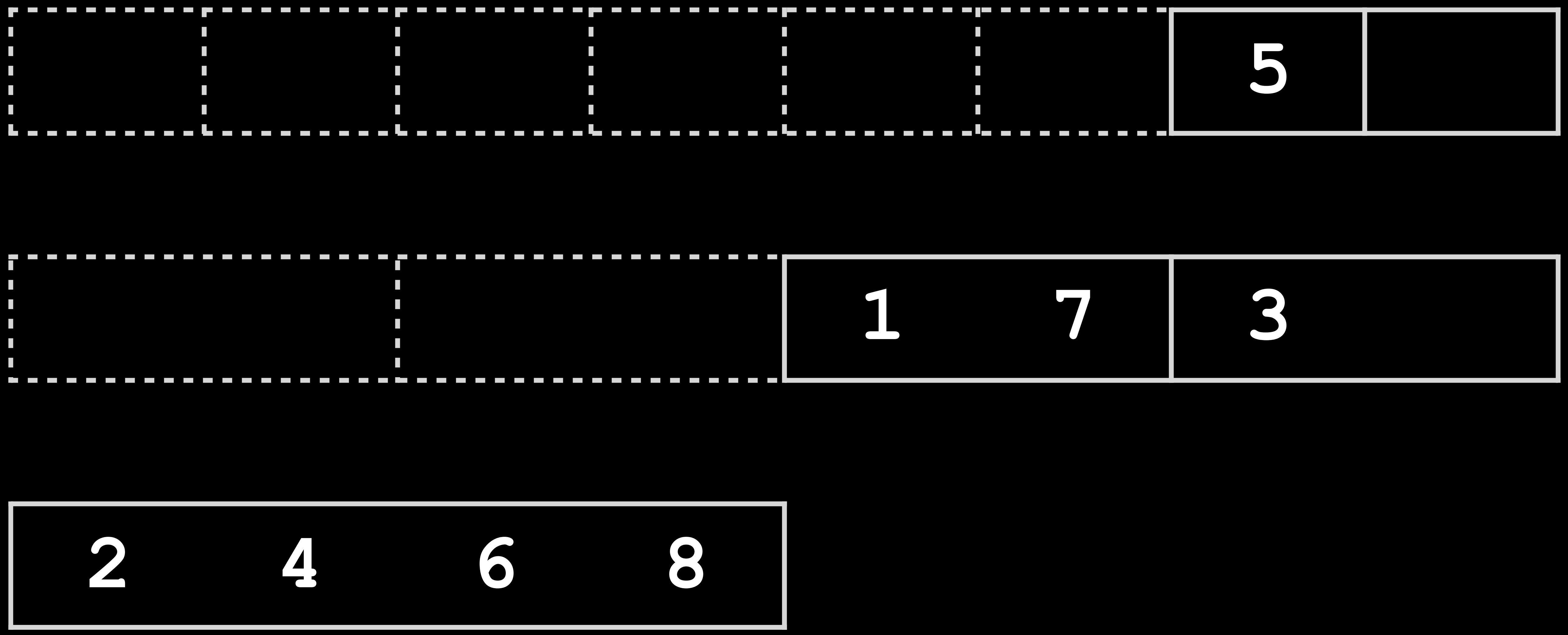
5 3

1 7

2 4 6 8


5 3
1 7
2 4 6 8

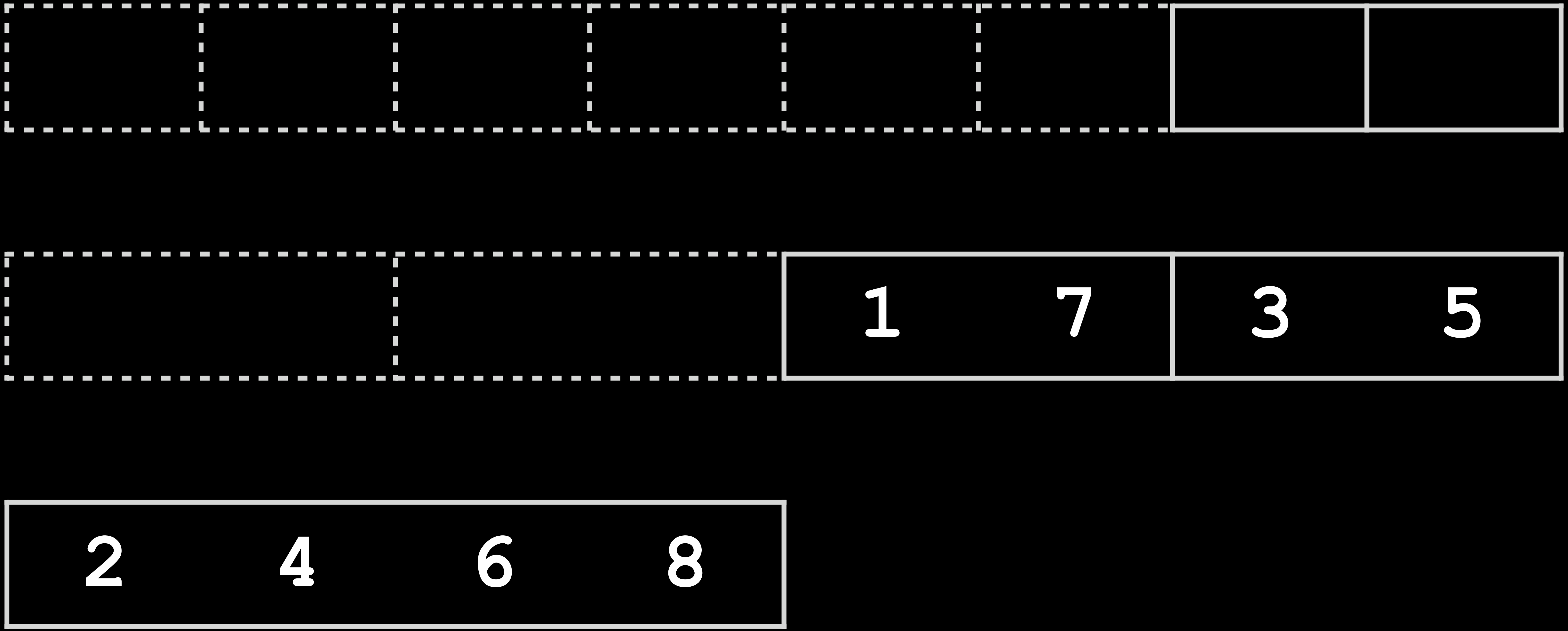

5
3
1 7
2 4 6 8



5

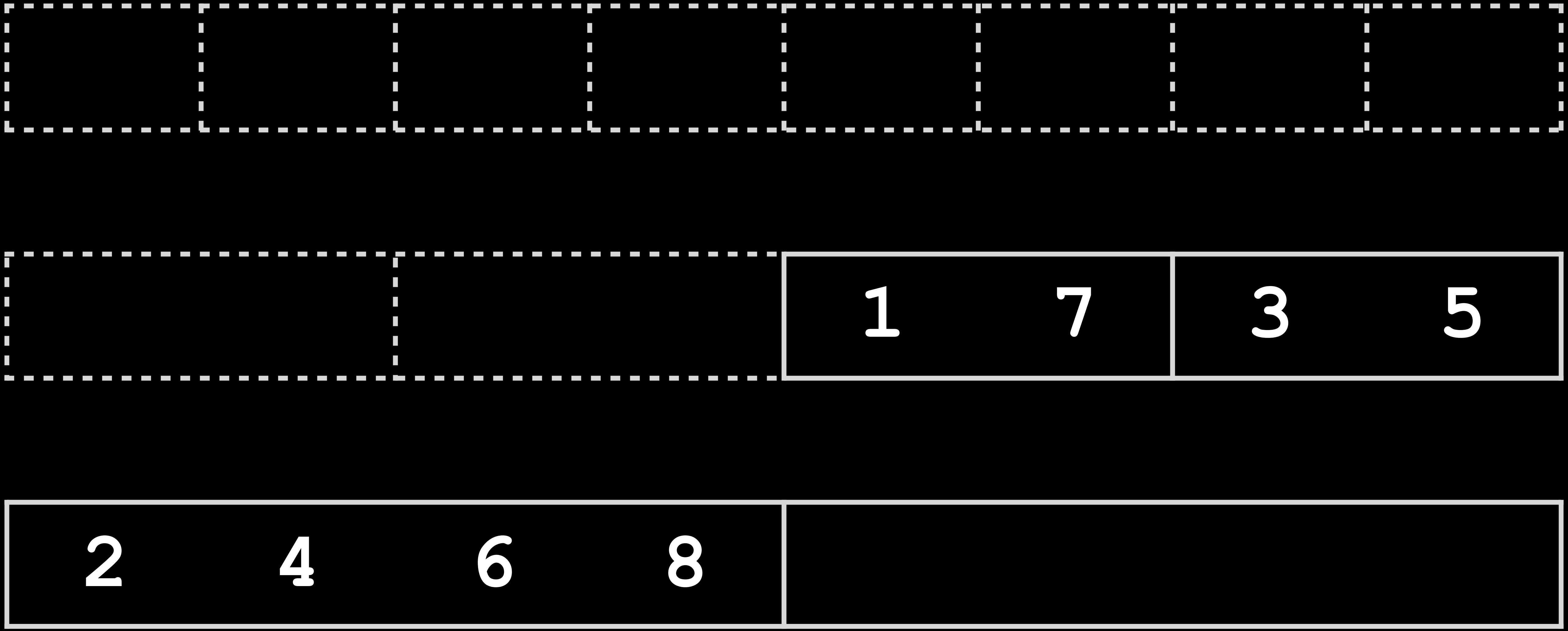
1 7 3

2 4 6 8



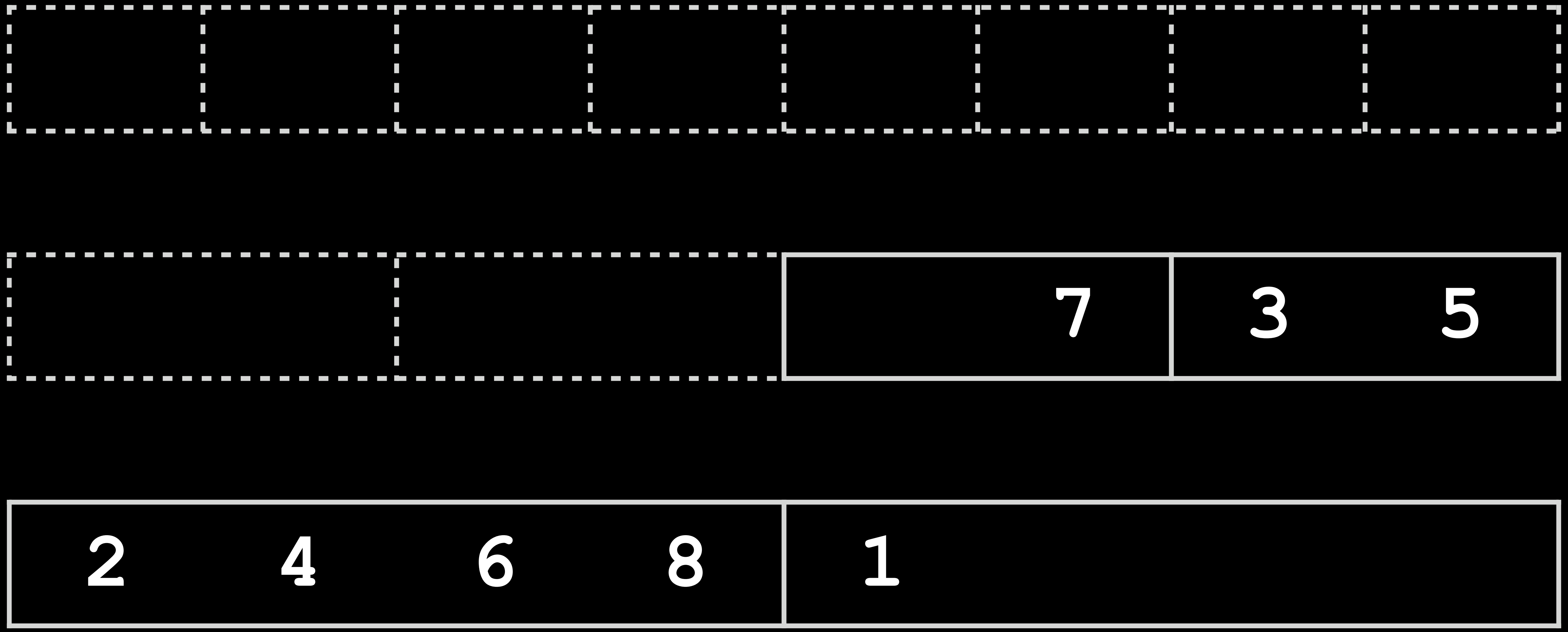
1 7 3 5

2 4 6 8



1 7 | 3 5

2 4 6 8 |

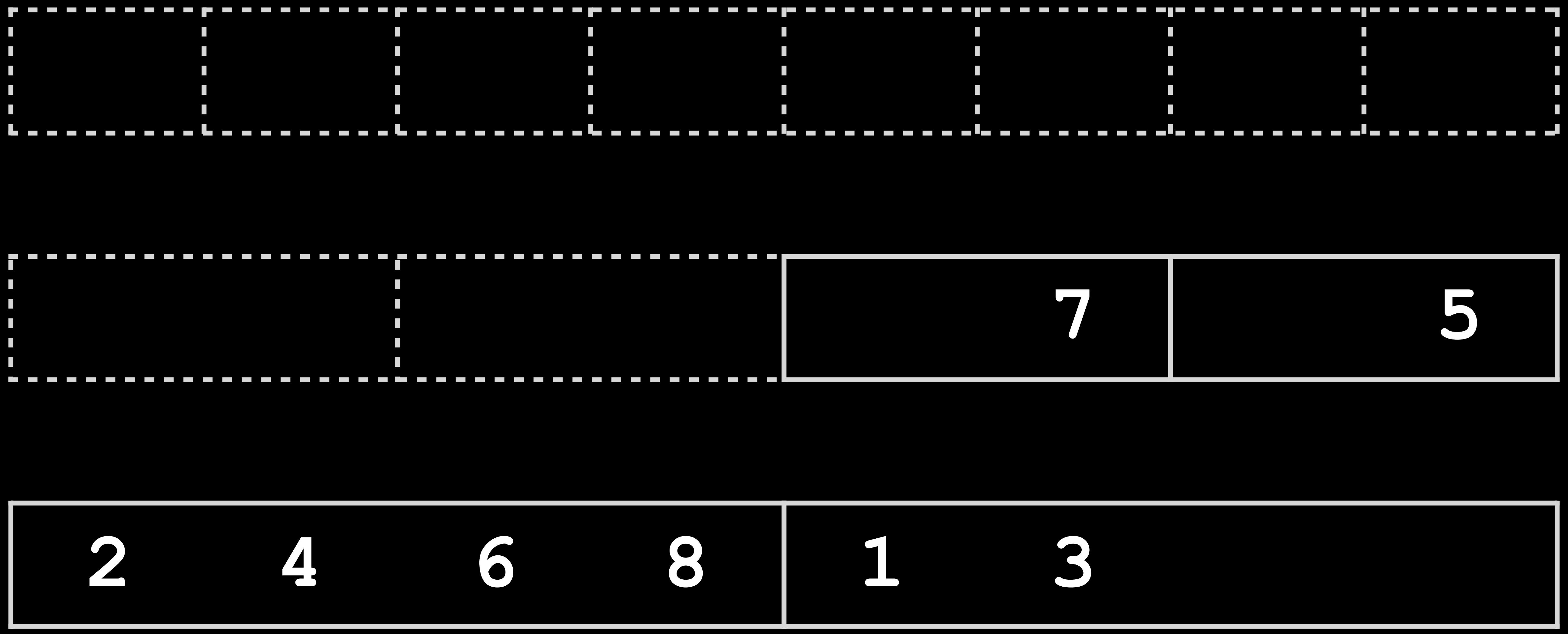


7

3 5

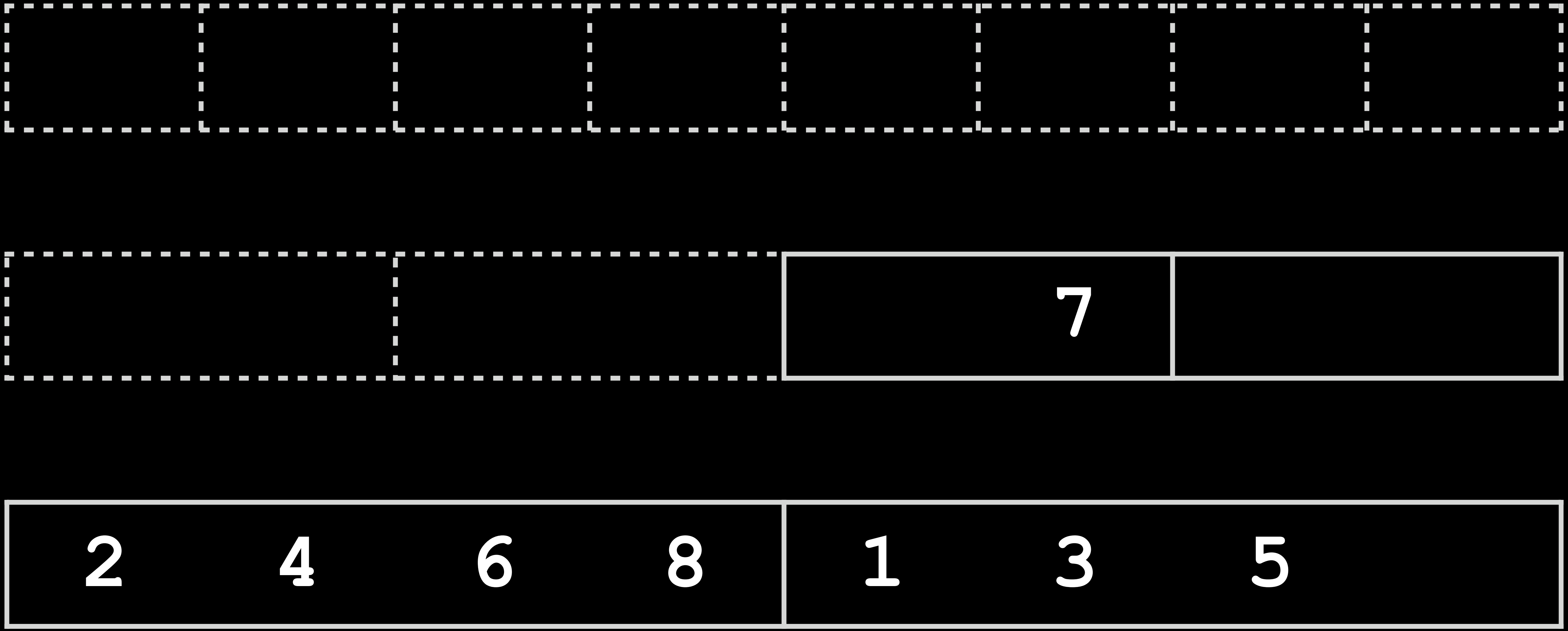
2 4 6 8

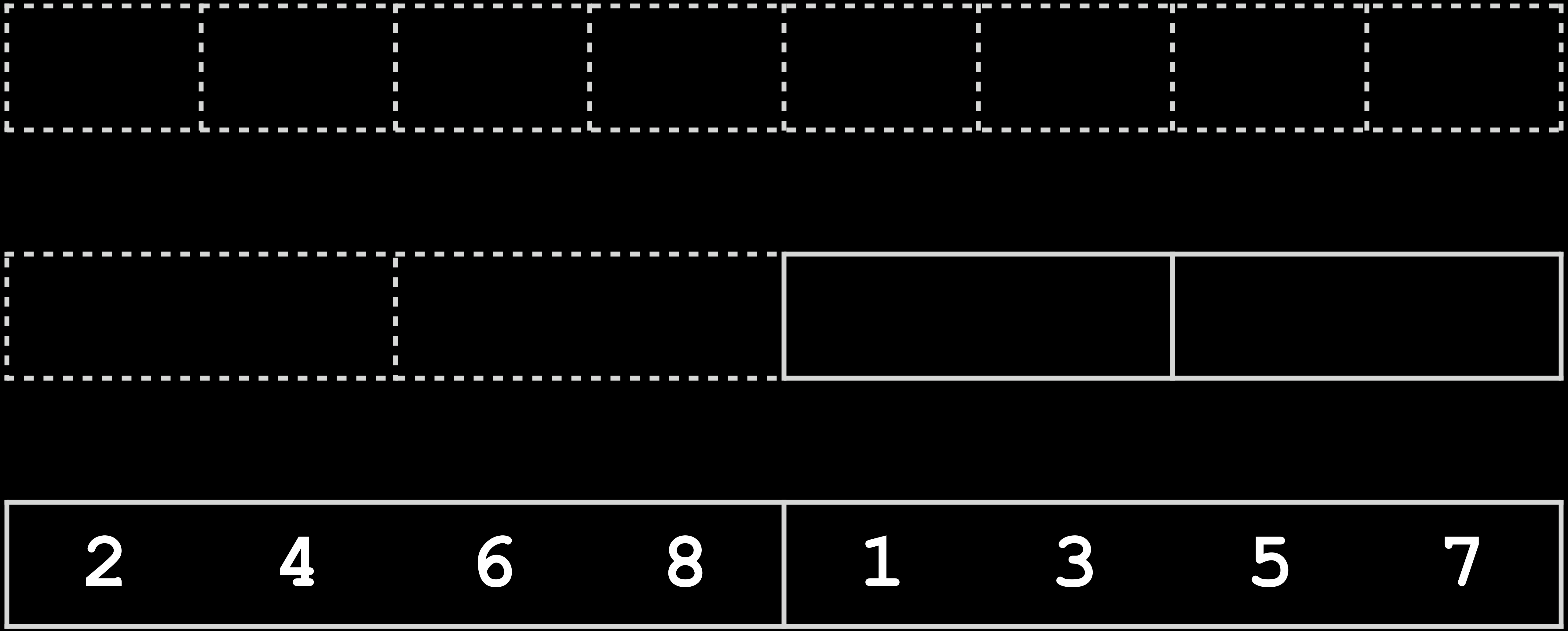
1



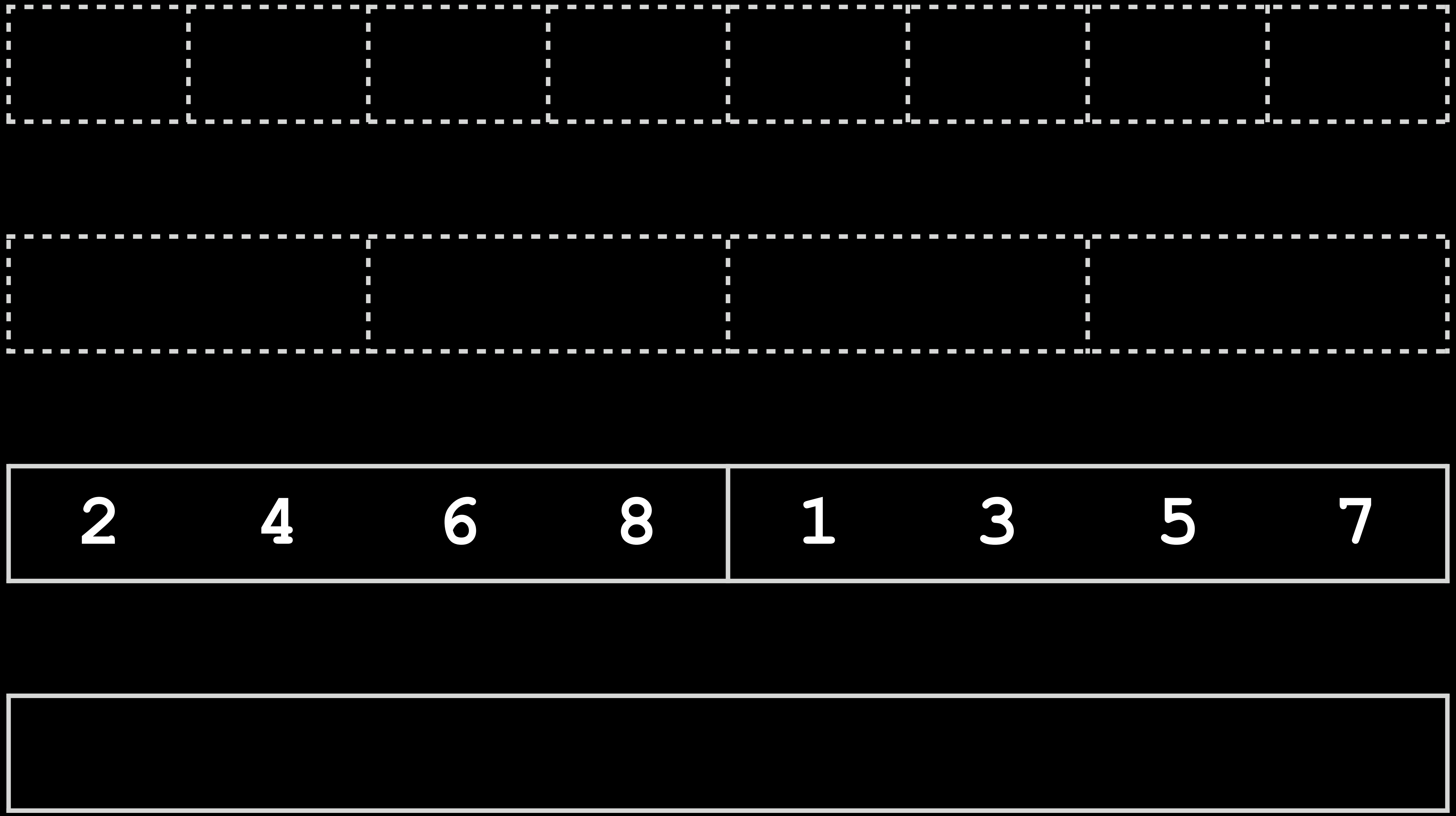
7 5

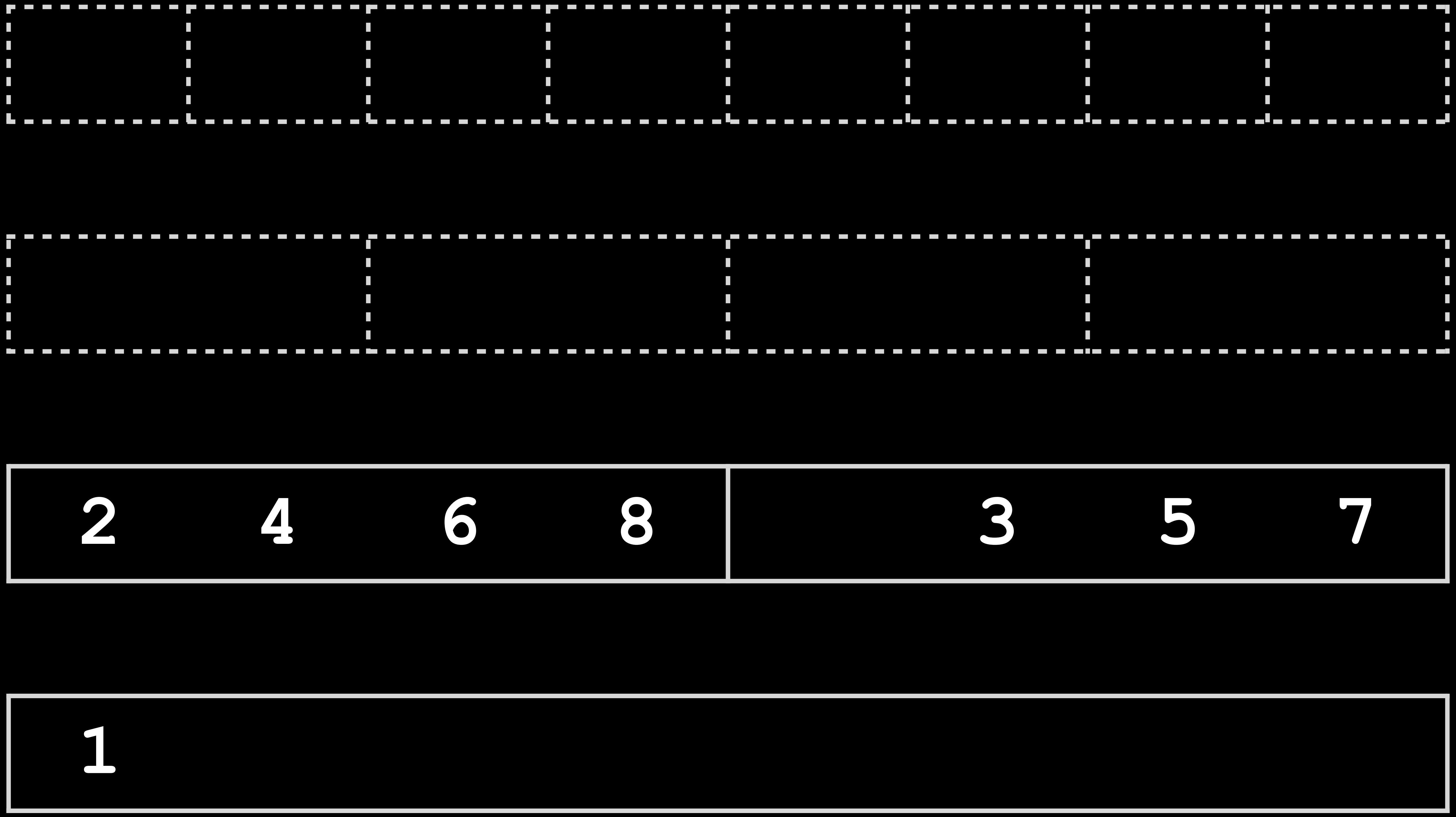
2 4 6 8 1 3


7
2 4 6 8
1 3 5



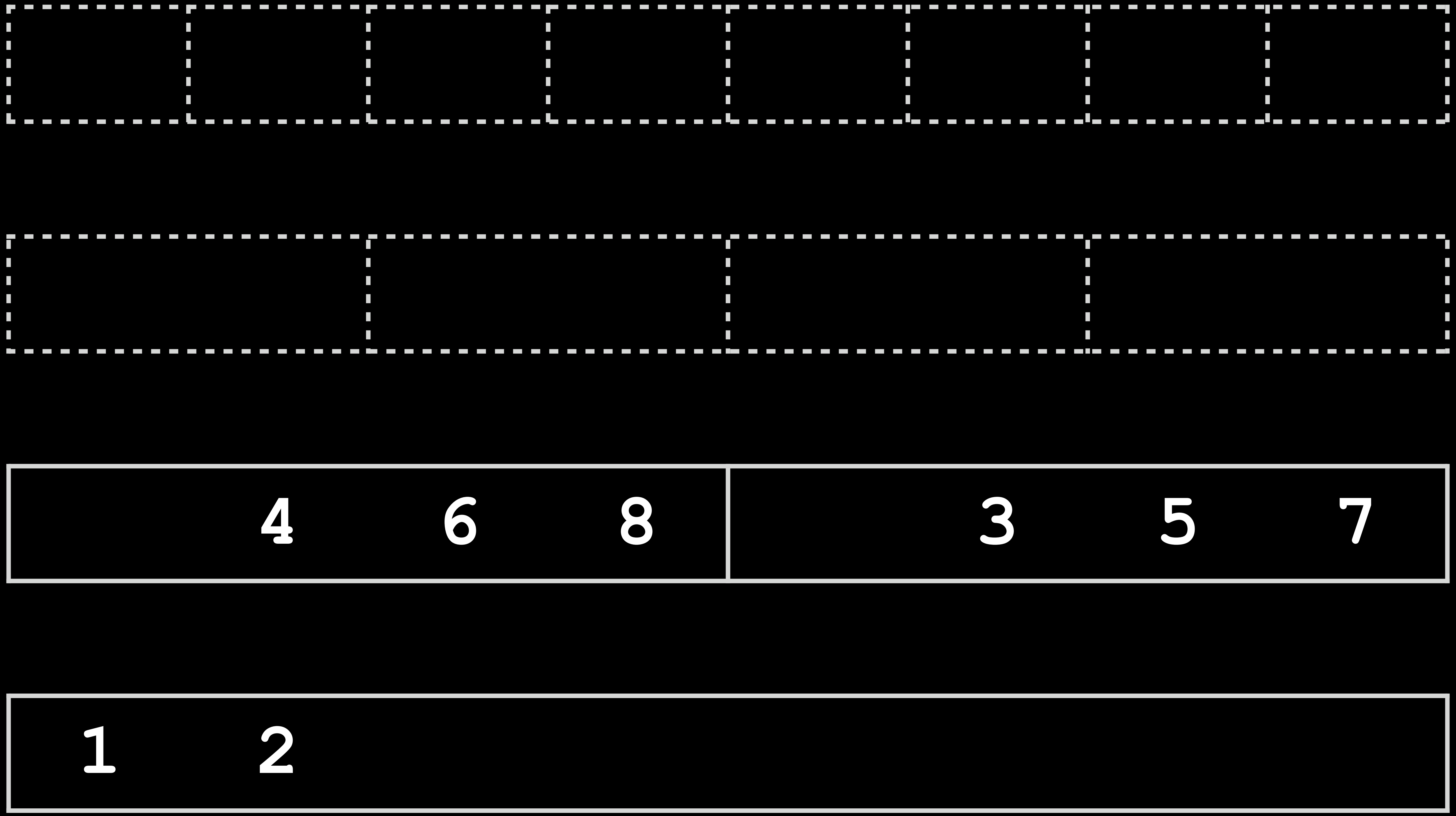
2 4 6 8 1 3 5 7


2
4
6
8
1
3
5
7



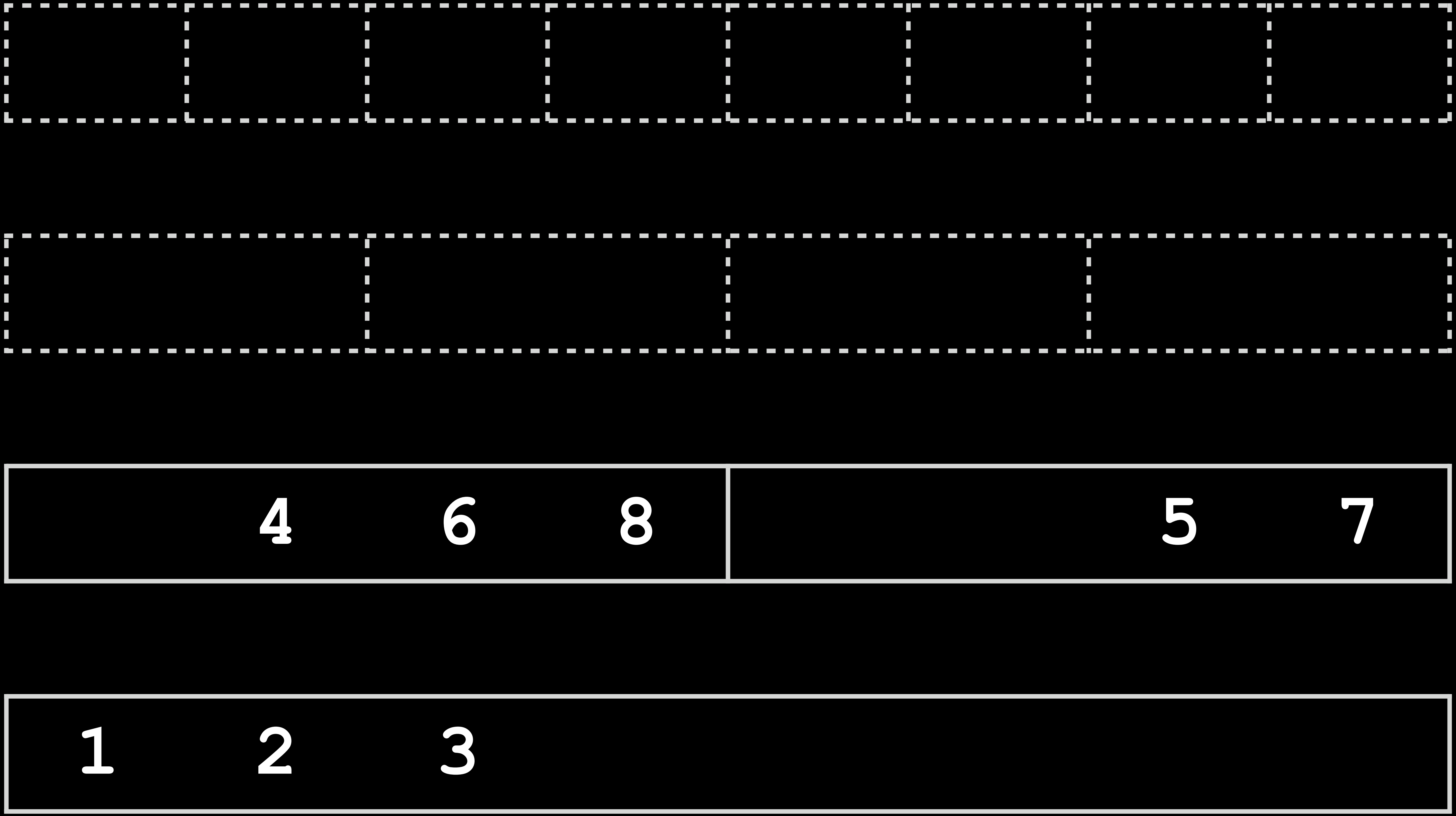
2 4 6 8 | 3 5 7

1



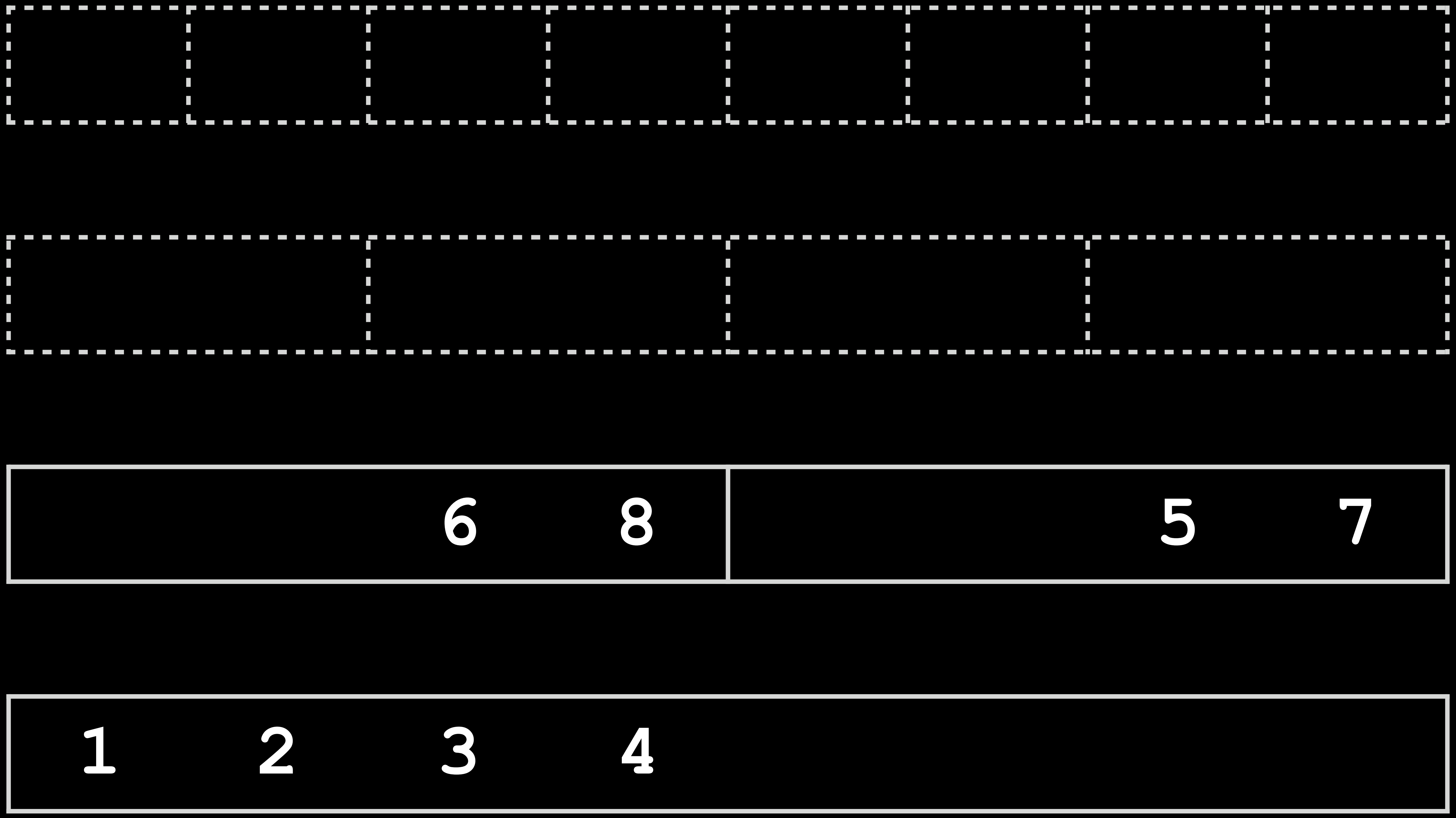
4 6 8 3 5 7

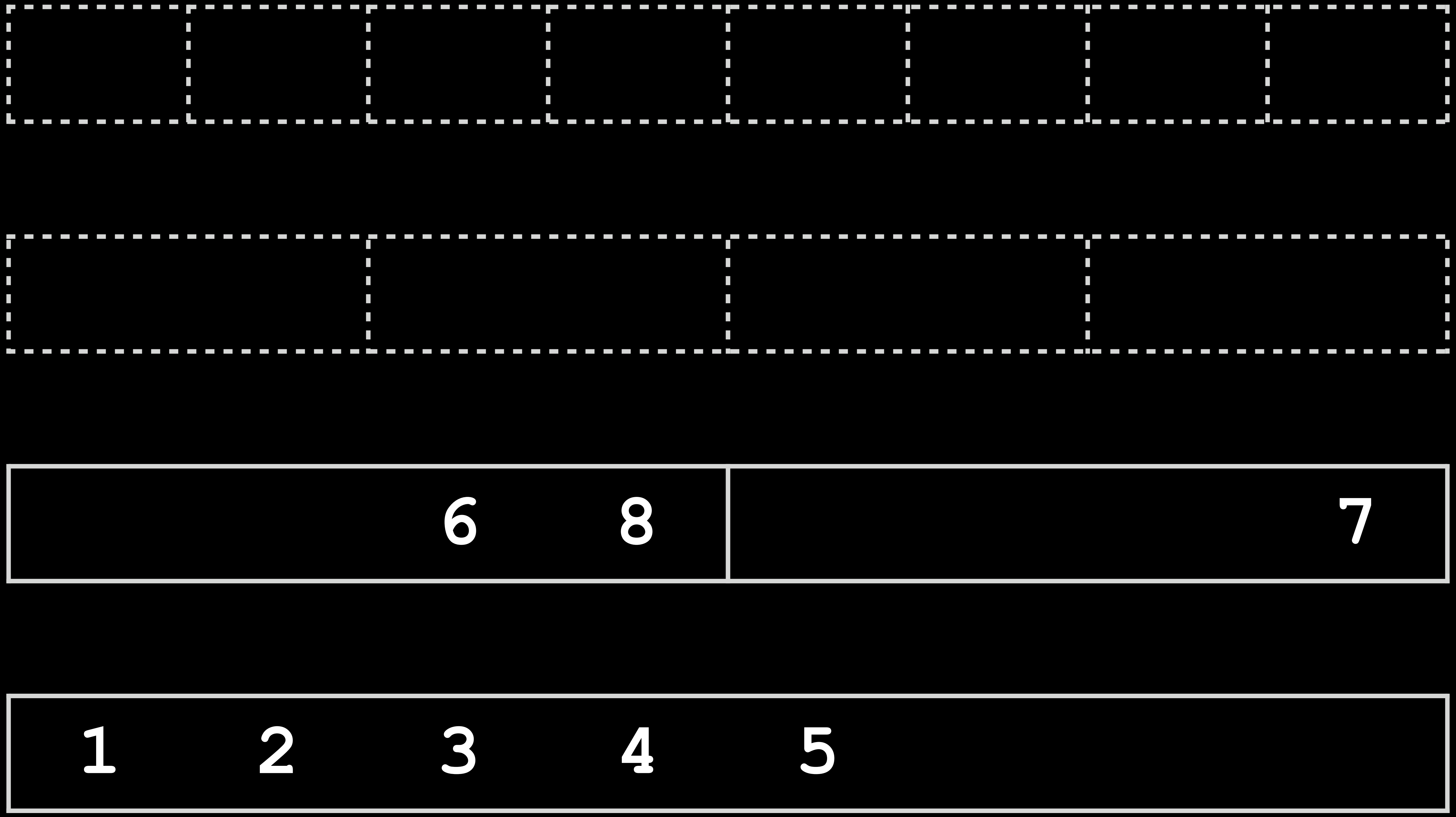
1 2

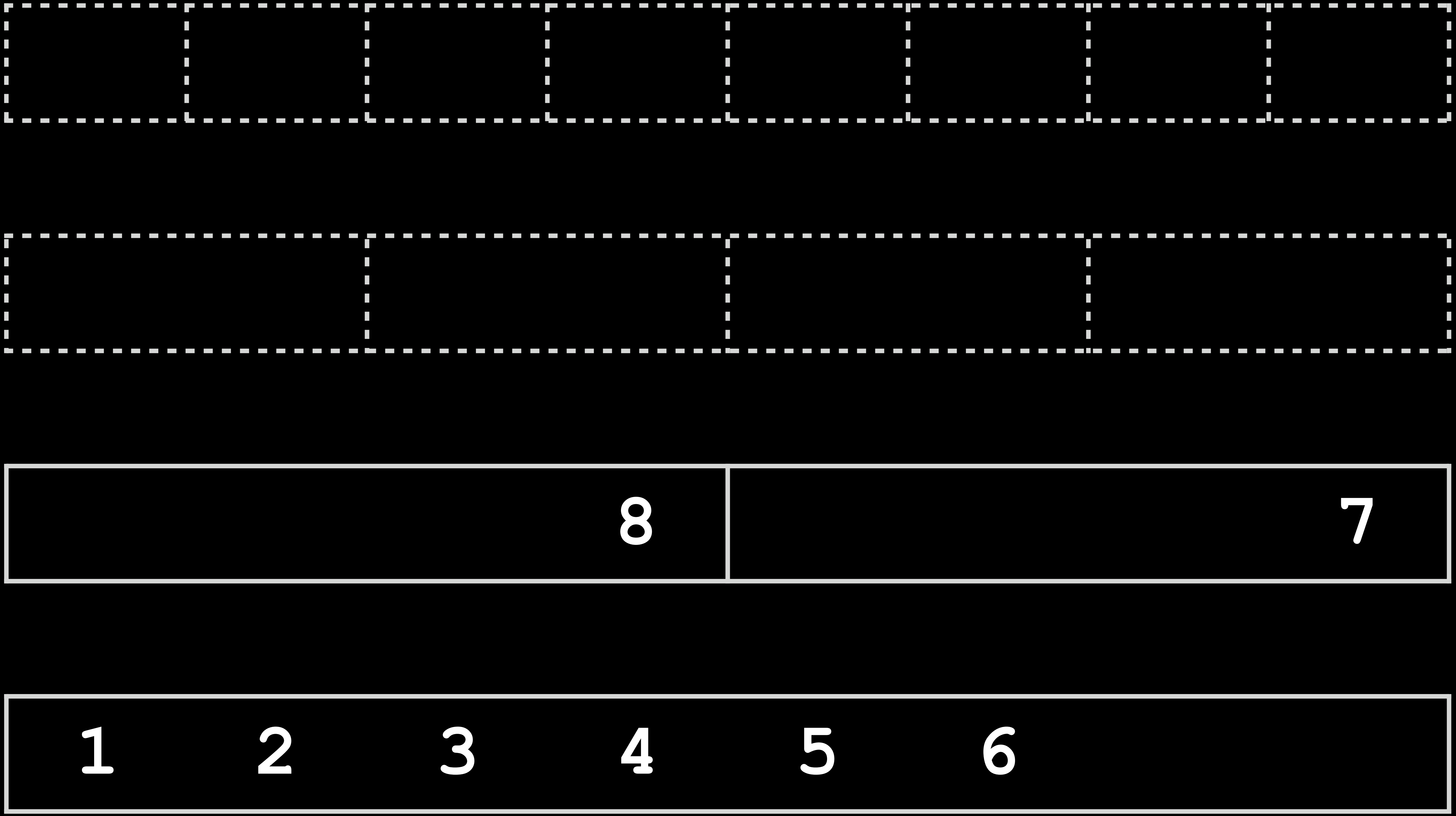


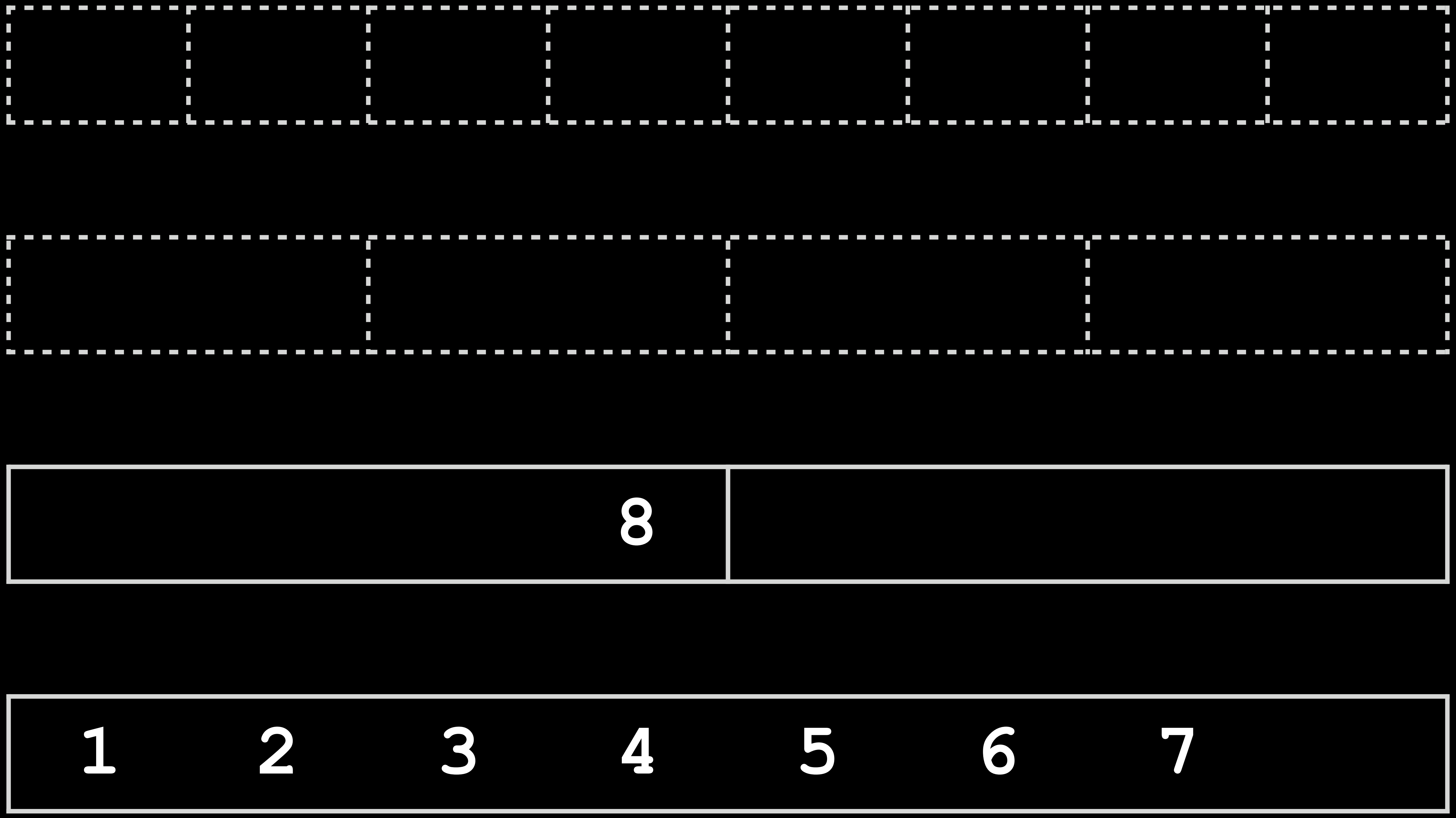
4 6 8 | 5 7

1 2 3


6 8
5 7
1 2 3 4

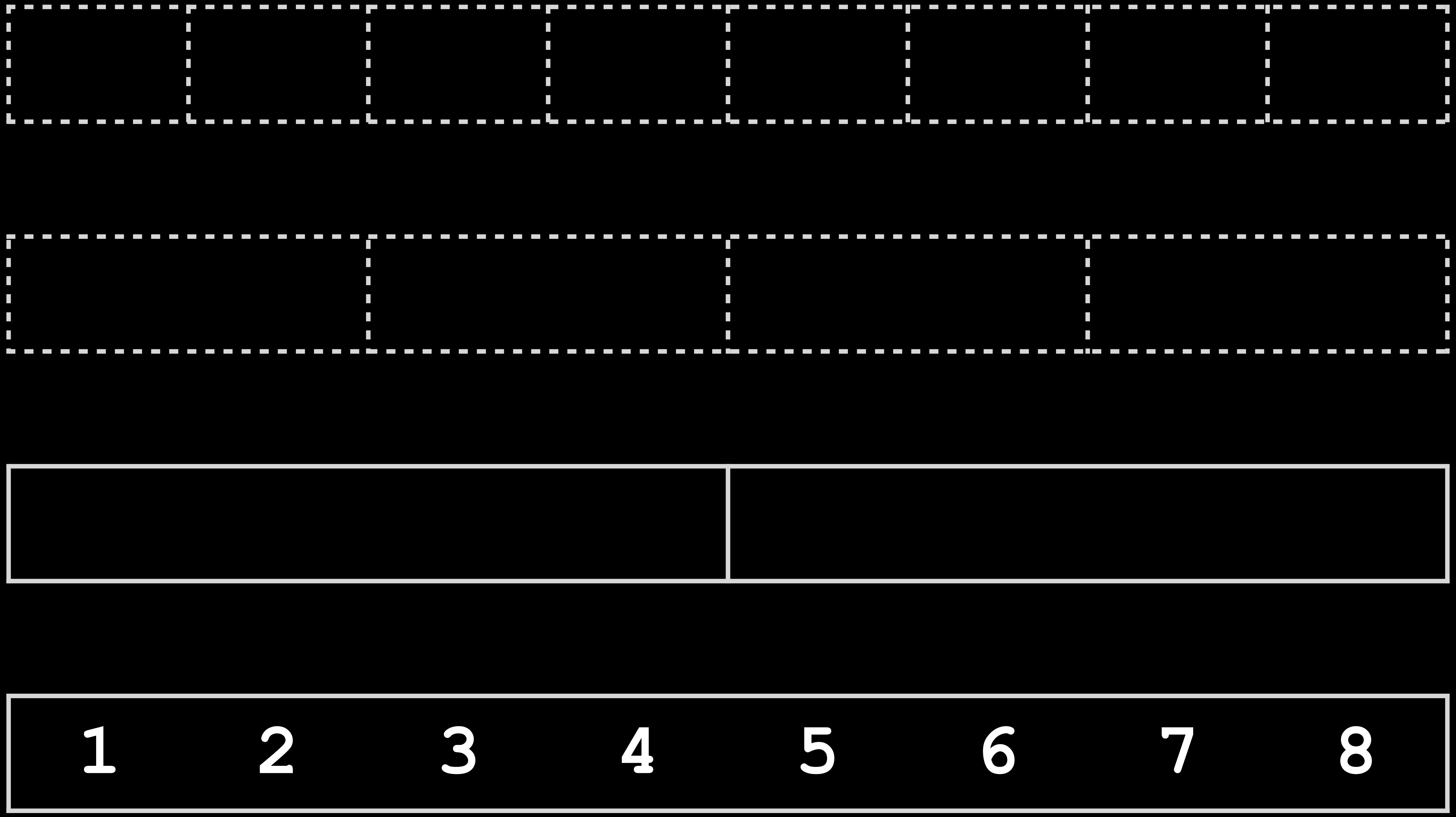

6 8
7
1 2 3 4 5


8
7
1
2
3
4
5
6

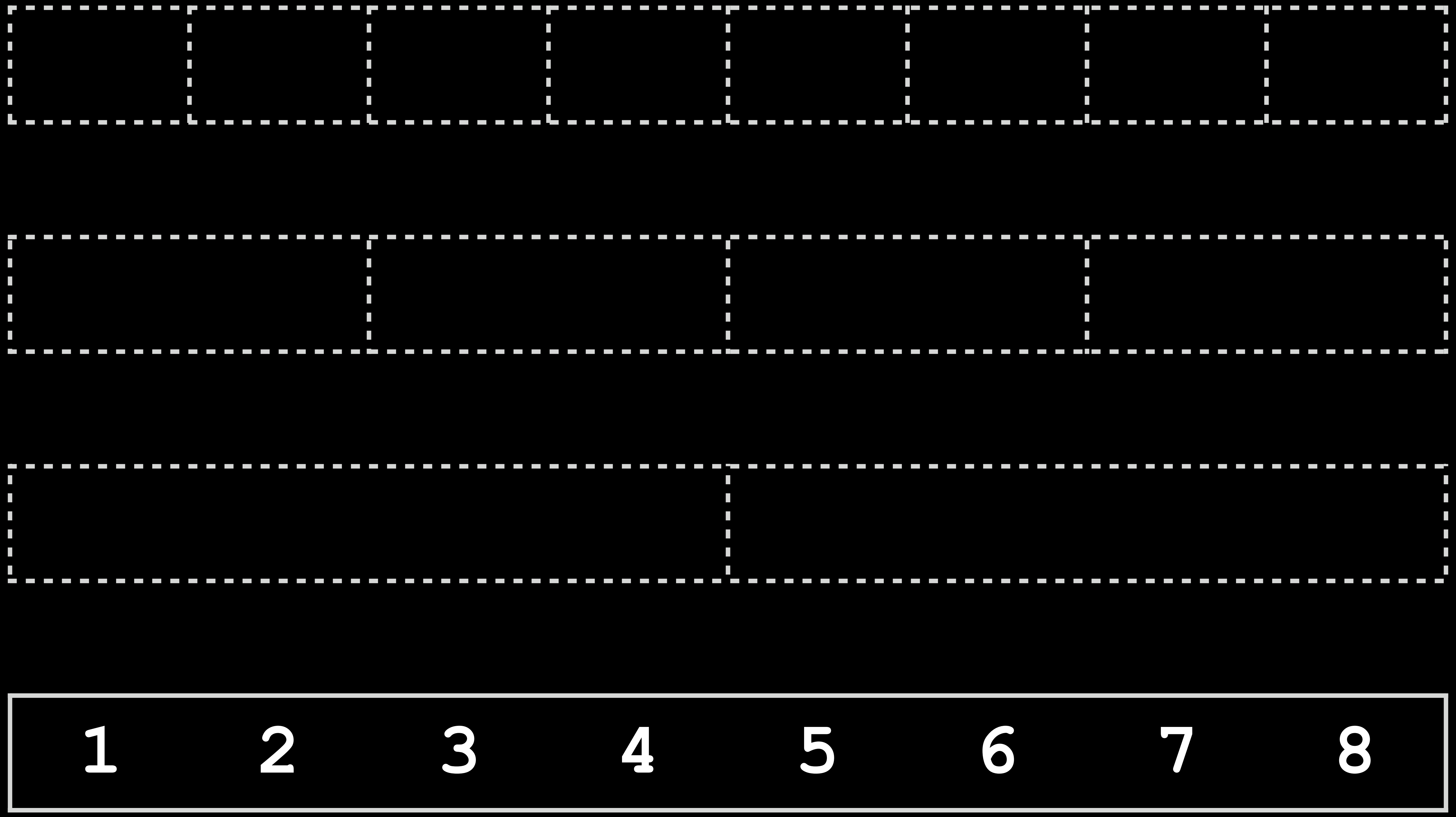


8

1 2 3 4 5 6 7



1 2 3 4 5 6 7 8



1 2 3 4 5 6 7 8

$O(n \log n)$

```
On input of $n$ elements
    if $n < 2$
        return
    else
        sort left half of elements
        sort right half of elements
        merge sorted halves
```

```
On input of $n$ elements
    if $n < 2$
        return
    else
        sort left half of elements
        sort right half of elements
        merge sorted halves
```

$$T(n) = O(1)$$

if $n < 2$

```
On input of $n$ elements
    if $n < 2$
        return
    else
        sort left half of elements
        sort right half of elements
        merge sorted halves
```

```
On input of $n$ elements
    if $n$ < 2
        return
    else
        sort left half of elements
        sort right half of elements
        merge sorted halves
```

```
On input of $n$ elements
    if $n < 2$
        return
    else
        sort left half of elements
        sort right half of elements
        merge sorted halves
```

$$T(n) = T(n/2) + T(n/2) + O(n)$$

if $n \geq 2$

$$O(n \log n)$$

Goo

Week 3