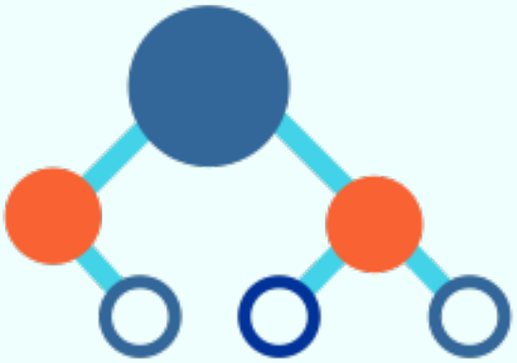
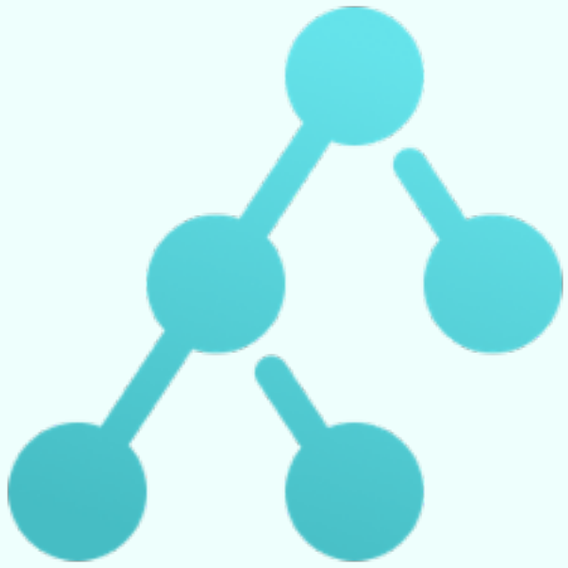
Data Structures

Exercise Session



Marco Vassena



Problem 14

```
quickSort(List xs)
  p = choosePivot(xs)
  S₁ = {x | x ∈ xs, x ≤ p}
  S₂ = {x | x ∈ xs, x > p}
  return  (quickSort S₁)  ++
          [ p ]           ++
          (quickSort S₂)
```

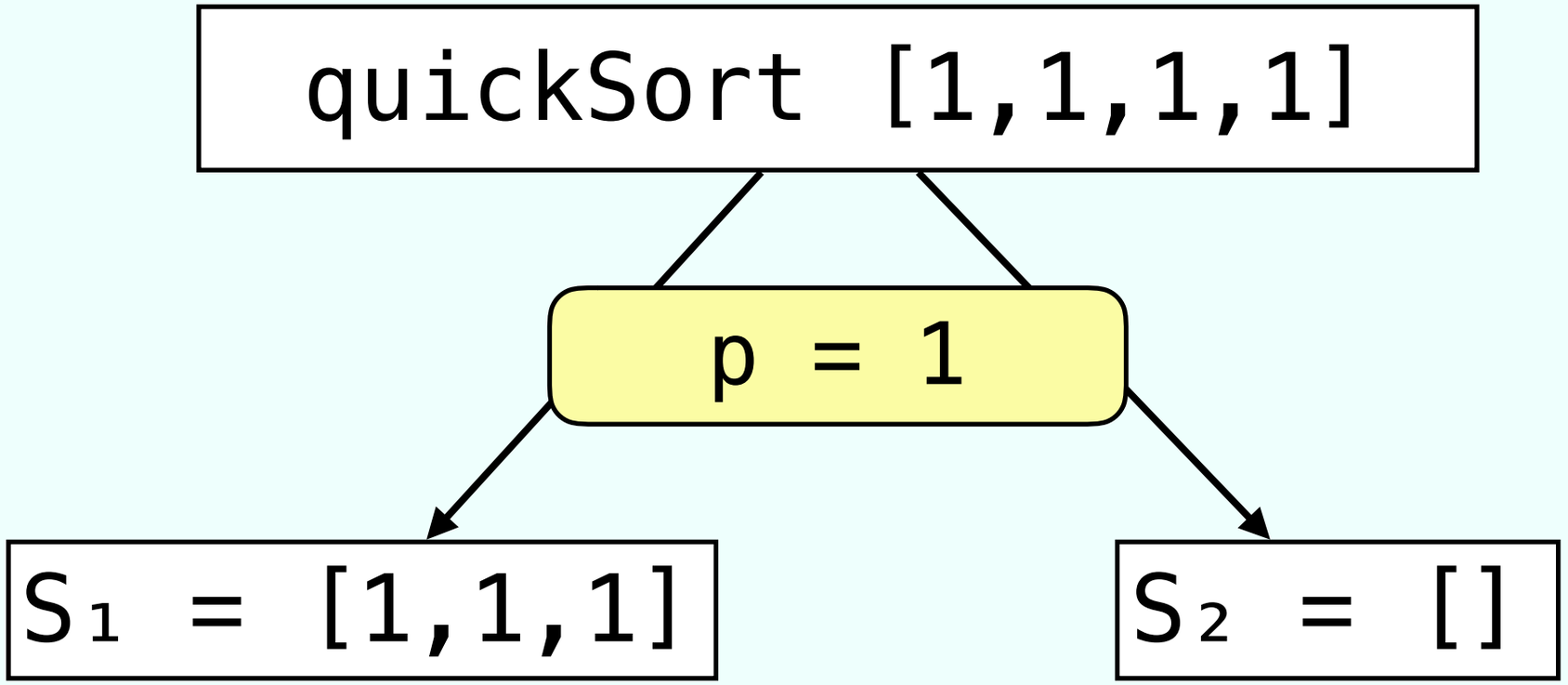
Problem 14

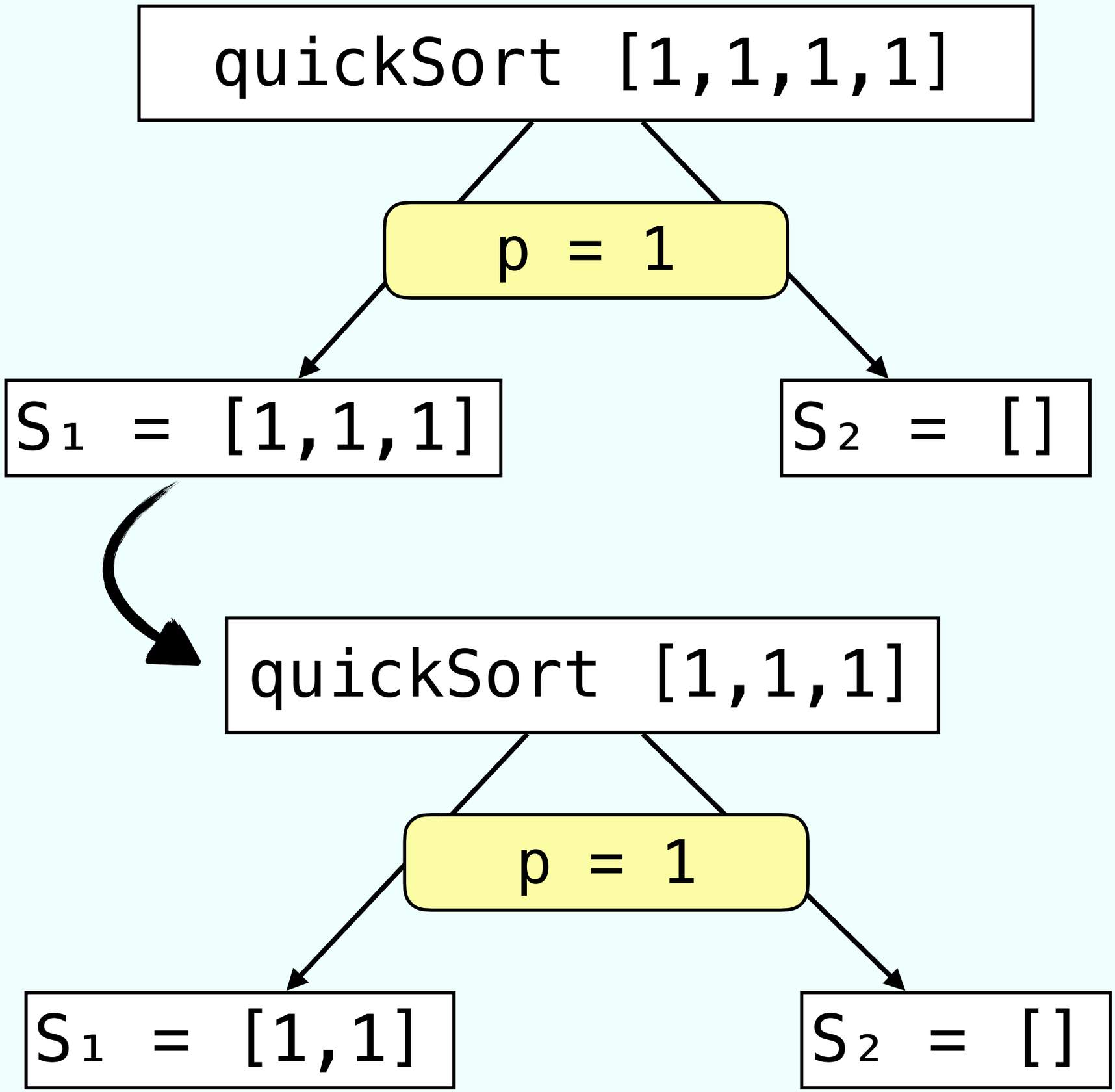
```
quickSort(List xs)
  p = choosePivot(xs)
  S₁ = {x | x ∈ xs, x ≤ p}
  S₂ = {x | x ∈ xs, x > p}
  return  (quickSort S₁)  ++
          [ p ]           ++
          (quickSort S₂)
```

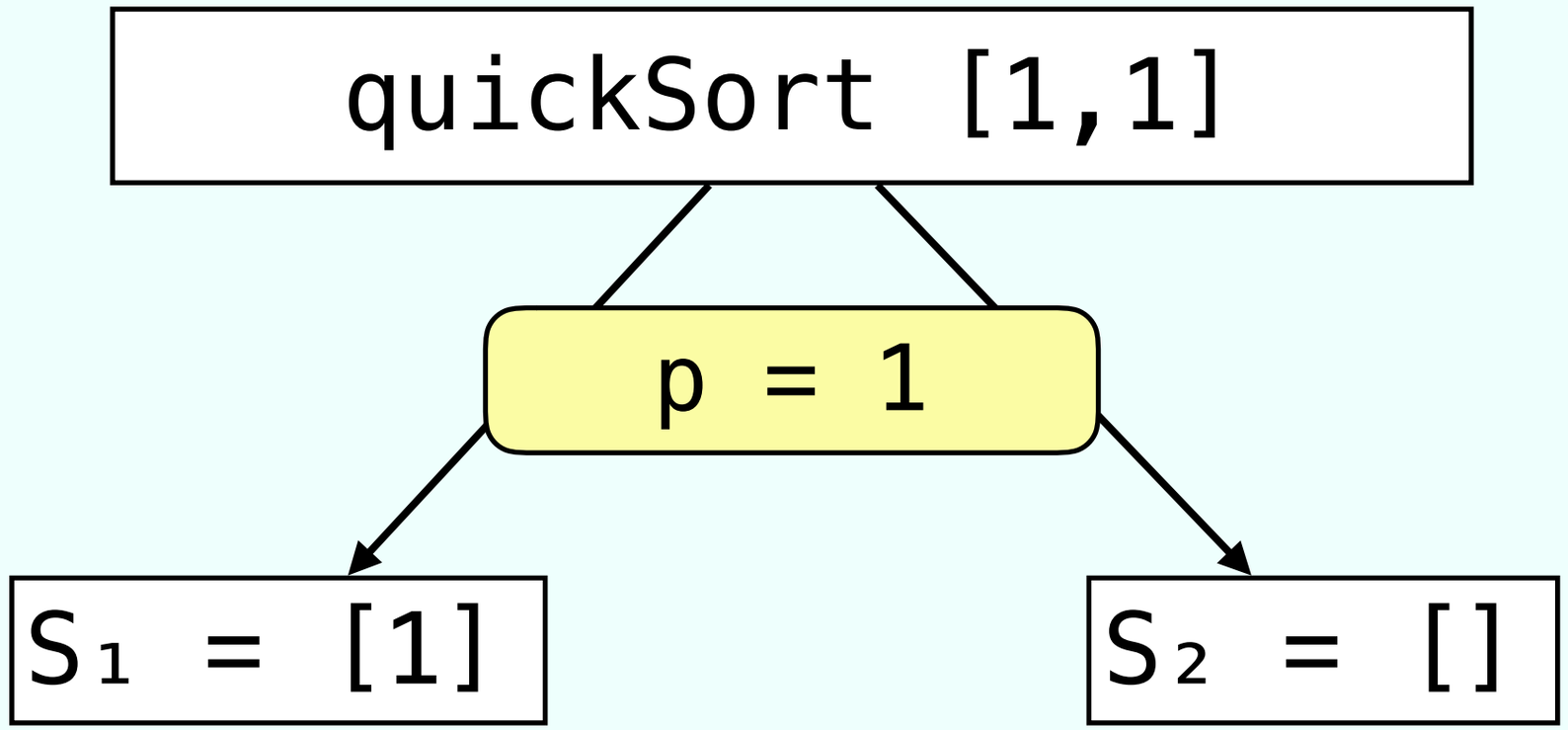
Problem 14

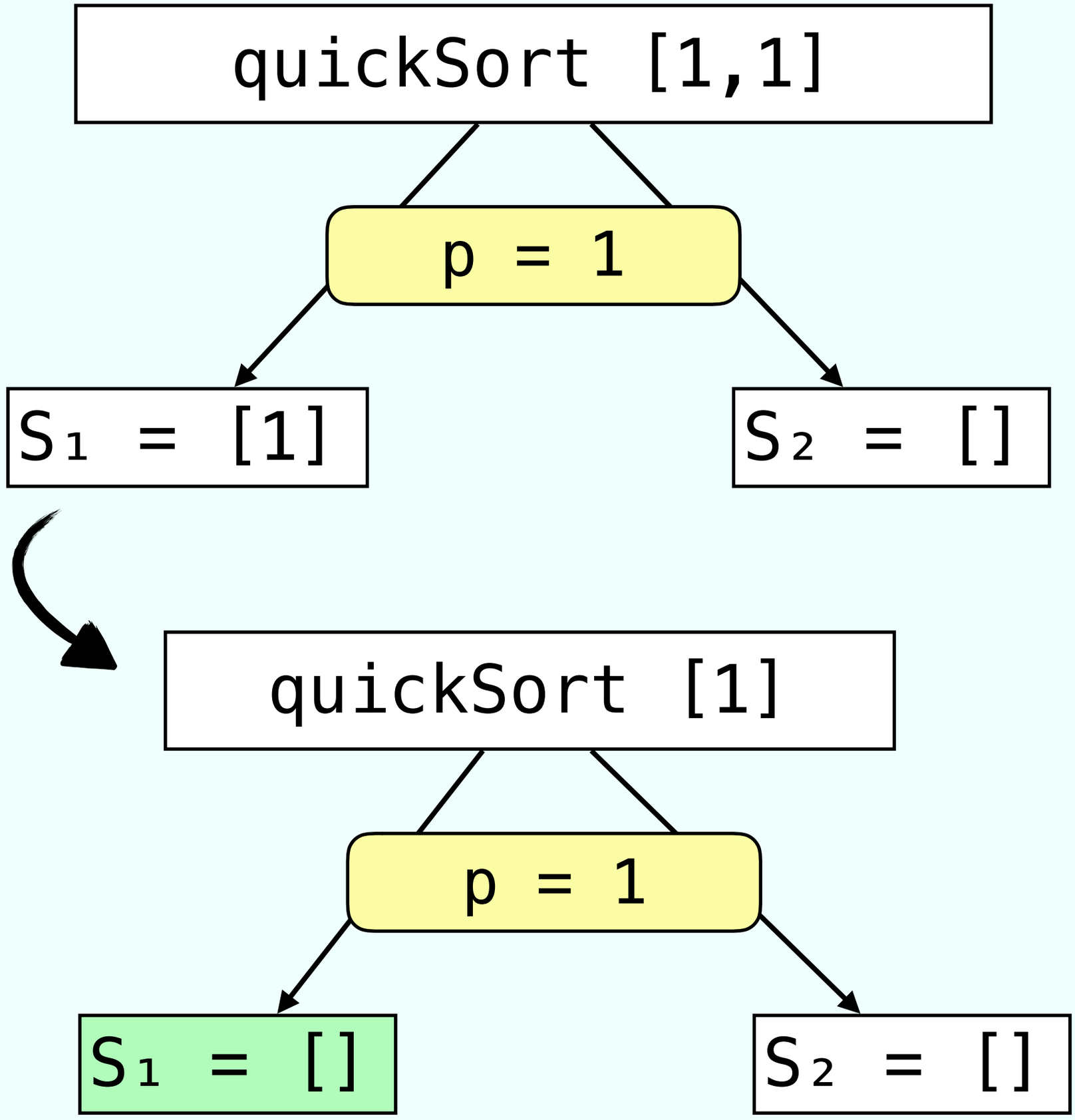
```
quickSort(List xs)
  p = choosePivot(xs)
  S₁ = {x | x ∈ xs, x ≤ p}
  S₂ = {x | x ∈ xs, x > p}
  return  (quickSort S₁)  ++
          [ p ]           ++
          (quickSort S₂)
```

Partition


quickSort [1,1,1,1]
p = 1
S1 = [1,1,1]
S2 = []


quickSort [1,1,1,1]
p = 1
S1 = [1,1,1]
S2 = []
quickSort [1,1,1]
p = 1
S1 = [1,1]
S2 = []


quickSort [1,1]
p = 1
S₁ = [1]
S₂ = []


quickSort [1,1]
p = 1
S1 = [1]
S2 = []
quickSort [1]
p = 1
S1 = []
S2 = []

```
quickSort(List xs)
```

```
p = choosePivot(xs)
S₁ = {x | x ∈ xs, x ≤ p}
S₂ = {x | x ∈ xs, x > p}
```

```
. . .
```

```
quickSort(List xs)
```

```
p = choosePivot(xs)
S₁ = {x | x ∈ xs, x ≤ p}
S₂ = {x | x ∈ xs, x > p}
```

```
...
```

$$|S_1| \approx |S_2| \approx O(N/2)$$

```
quickSort(List xs)
```

```
p = choosePivot(xs)
S₁ = {x | x ∈ xs, x ≤ p}
S₂ = {x | x ∈ xs, x > p}
```

```
...
```

$|S_1| \approx |S_2| \approx 0(N/2)$

$O(N \log N)$

```
quickSort(List xs)
```

```
p = choosePivot(xs)
S₁ = {x | x ∈ xs, x ≤ p}
S₂ = {x | x ∈ xs, x > p}
```

```
. . .
```

```
quickSort(List xs)
```

```
p = choosePivot(xs)
S₁ = {x | x ∈ xs, x ≤ p}
S₂ = {x | x ∈ xs, x > p}
```

```
...
```

$$|S_1| \approx O(N) \quad \text{v} \quad |S_2| \approx O(N)$$

```
quickSort(List xs)
```

```
p = choosePivot(xs)
S₁ = {x | x ∈ xs, x ≤ p}
S₂ = {x | x ∈ xs, x > p}
...
```

$|S_1| \approx O(N) \quad v \quad |S_2| \approx O(N)$

$O(N^2)$

Exercise 4 from12/04

Compute the product of the k smallest numbers in an array.

Exercise 4 from12/04

Compute the product of the k smallest numbers in an array.

```
k-product([3,7,5,4], 0) = 1
```

Exercise 4 from12/04

Compute the product of the k smallest numbers in an array.

```
k-product([3,7,5,4], 0) = 1
```

```
k-product([3,7,5,4], 1) = 3
```

Exercise 4 from12/04

Compute the product of the k smallest numbers in an array.

```
k-product([3,7,5,4], 0) = 1
```

```
k-product([3,7,5,4], 1) = 3
```

```
k-product([3,7,5,4], 2) = 12
```

Exercise 4 from12/04

Compute the product of the k smallest numbers in an array.

```
k-product([3,7,5,4], 0) = 1
```

```
k-product([3,7,5,4], 1) = 3
```

```
k-product([3,7,5,4], 2) = 12
```

```
k-product([3,7,5,4], 3) = 60
```

Exercise 4 from12/04

Compute the product of the k smallest numbers in an array.

O(N)

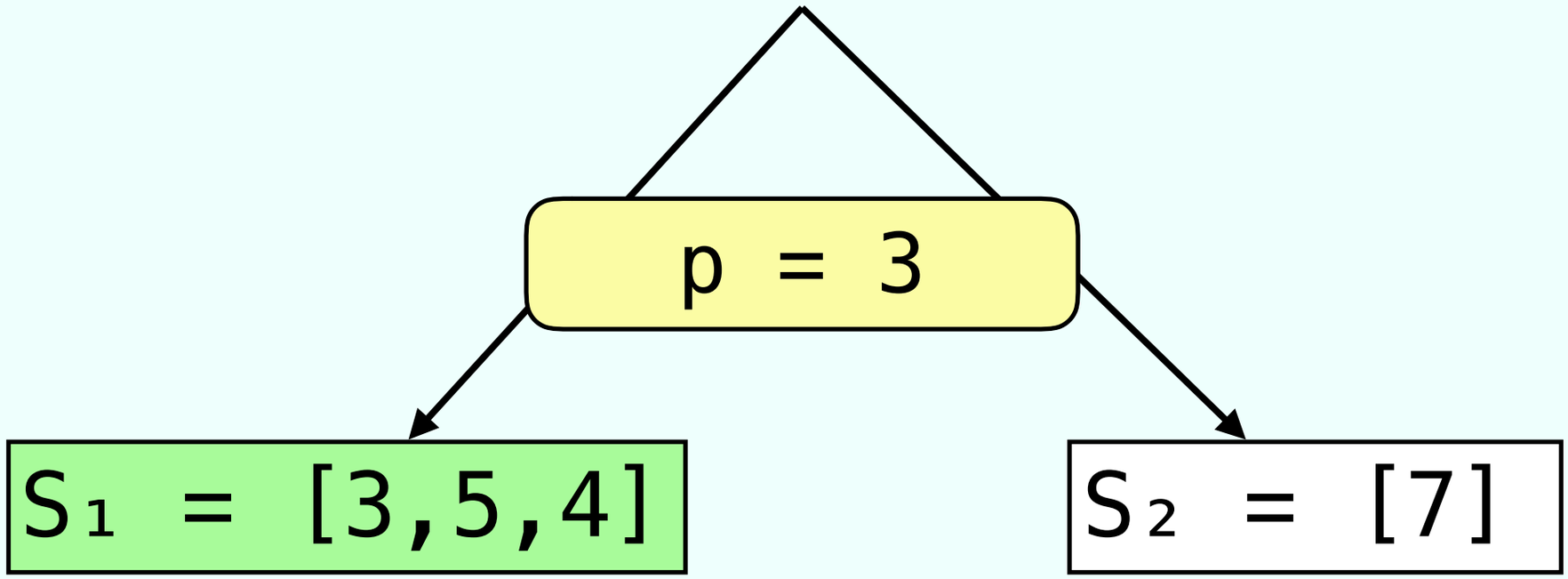
```
k-product([3,7,5,4], 0) = 1
```

```
k-product([3,7,5,4], 1) = 3
```

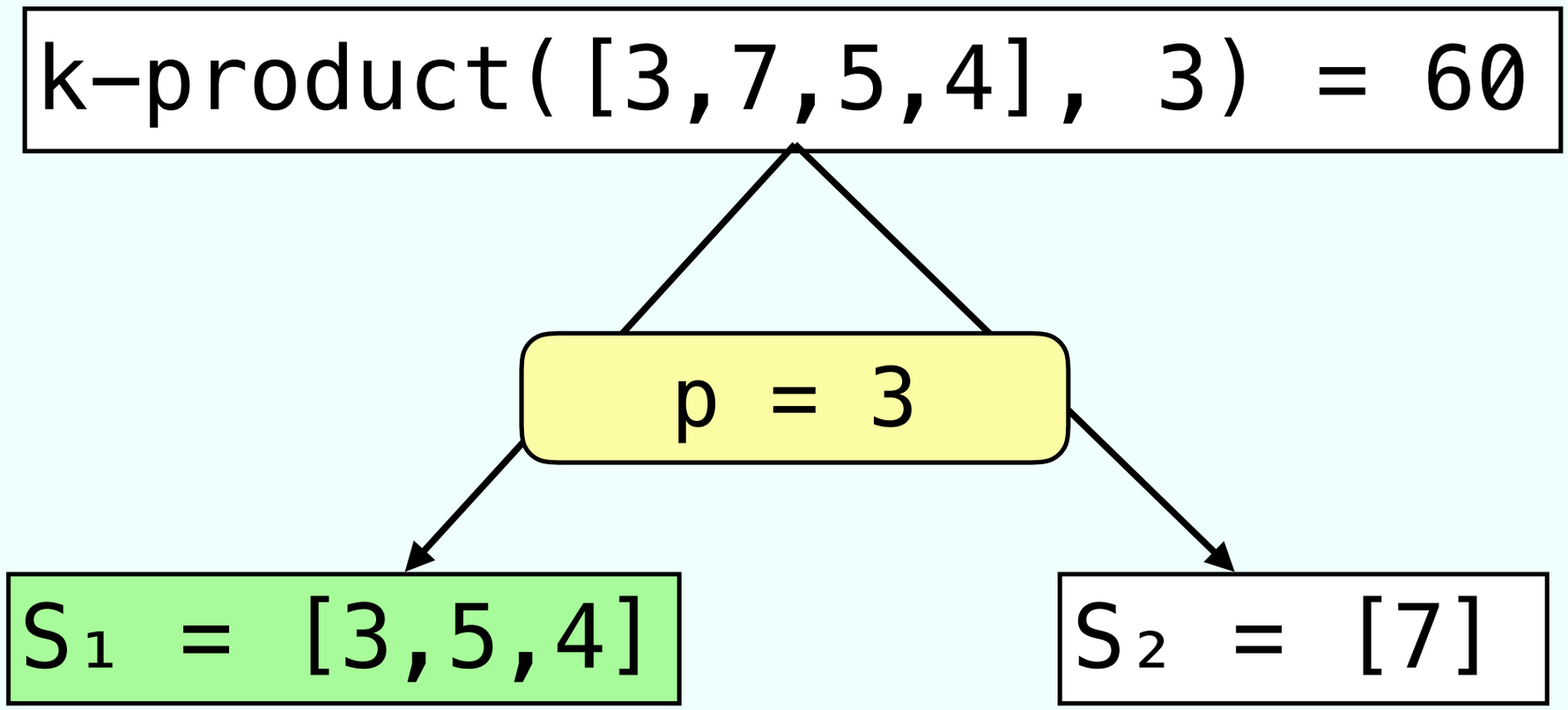
```
k-product([3,7,5,4], 2) = 12
```

```
k-product([3,7,5,4], 3) = 60
```

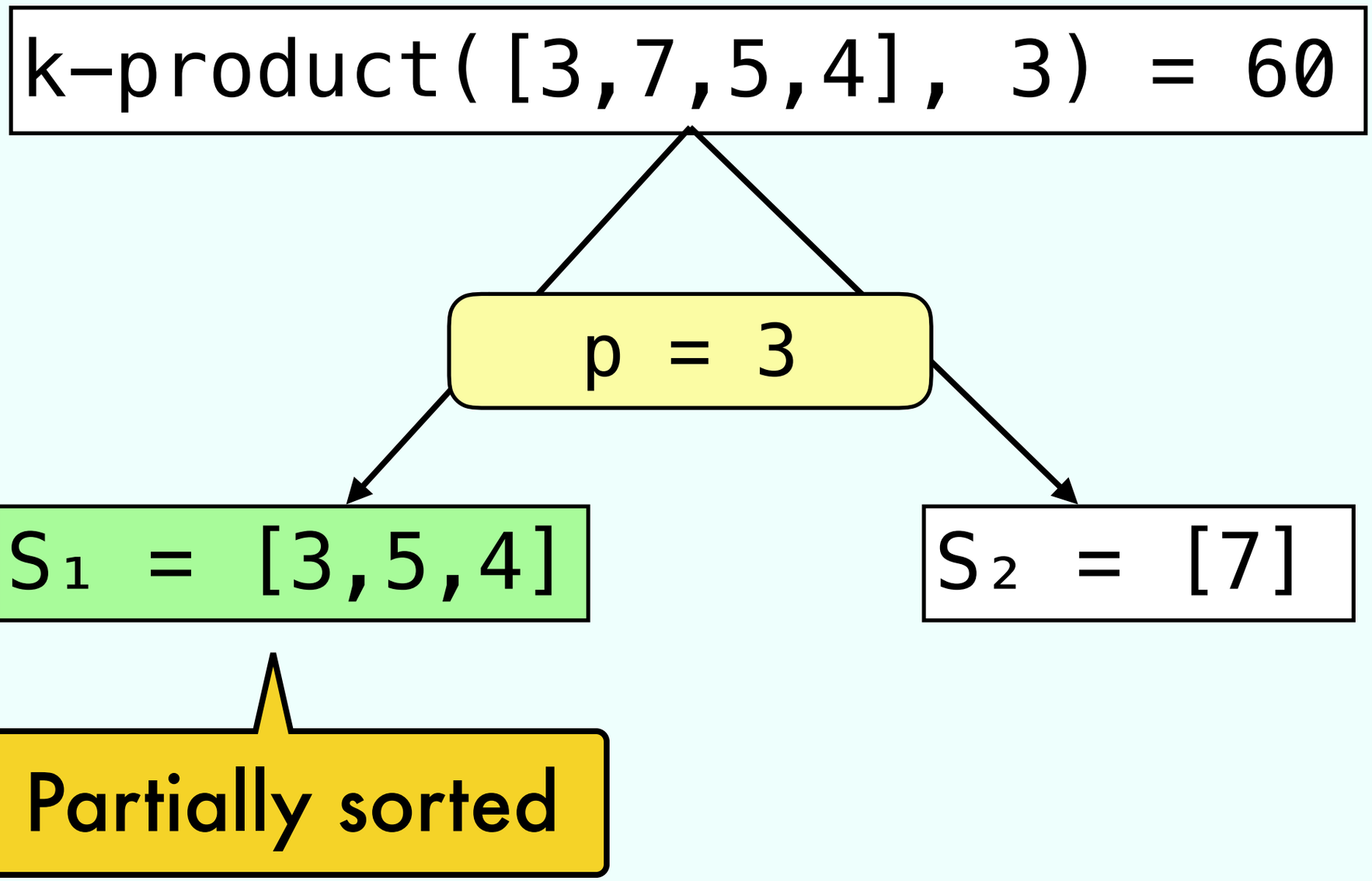
Insight



Insight



Insight



```
select-k-smallest(xs, L, R, k){
loop
  if L == R
    return xs[L]
  p = partition(xs, L, R)
  if k == p
    return xs[k]
  if k < p
    R = p - 1
  else
    L = p + 1
}
```

```
select-k-smallest(xs, L, R, k){
loop
  if L == R
    return xs[L]
  p = partition(xs, L, R)
  if k == p
    return xs[k]
  if k < p
    R = p - 1
  else
    L = p + 1
}
```

L ≤ k ≤ M

```
select-k-smallest(xs, L, R, k){
loop
  if L == R
    return xs[L]
  p = partition(xs, L, R)
  if k == p
    return xs[k]
  if k < p
    R = p - 1
  else
    L = p + 1
}
```

L ≤ k ≤ M

i < p ⇒ xs[i] ≤ xs[p]
i > p ⇒ xs[i] > xs[p]

```
select-k-smallest-index(xs, L, R, k){
loop
  if L == R
    return L
  p = partition(xs, L, R)
  if k == p
    return k
  if k < p
    R = p - 1
  else
    L = p + 1
}
```

L ≤ k ≤ M

i < p ⇒ xs[i] ≤ xs[p]
i > p ⇒ xs[i] > xs[p]

Exercise 3 from 12/12

Does a list contain 2 equal numbers?

Exercise 3 from 12/12

Does a list contain 2 equal numbers?

O(N)

Exercise 3 from 12/12

Does a list contain 2 equal numbers?

O(N)

```
hasDouble([1,2,3,1]) =
```

Exercise 3 from 12/12

Does a list contain 2 equal numbers?

O(N)

```
hasDouble([1,2,3,1]) =
```

True

Exercise 3 from 12/12

Does a list contain 2 equal numbers?

O(N)

```
hasDouble([1,2,3,1]) =
```

True

```
hasDouble([1,2,3,4]) =
```

Exercise 3 from 12/12

Does a list contain 2 equal numbers?

O(N)

```
hasDouble([1,2,3,1]) = True
```

```
hasDouble([1,2,3,4]) = False
```