

123Freevectors.com

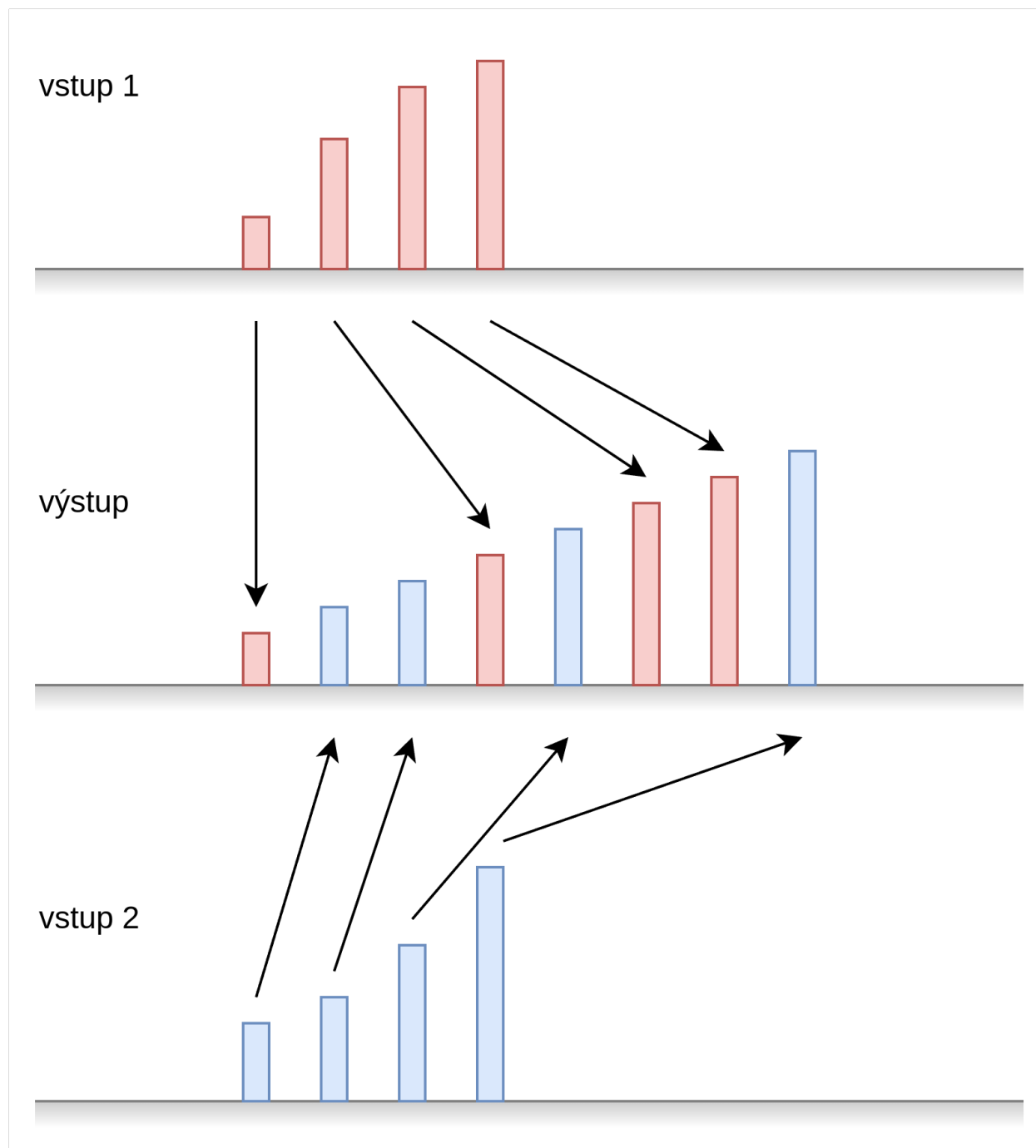
Programovacie techniky

3. prednáška, 4. cvičenie

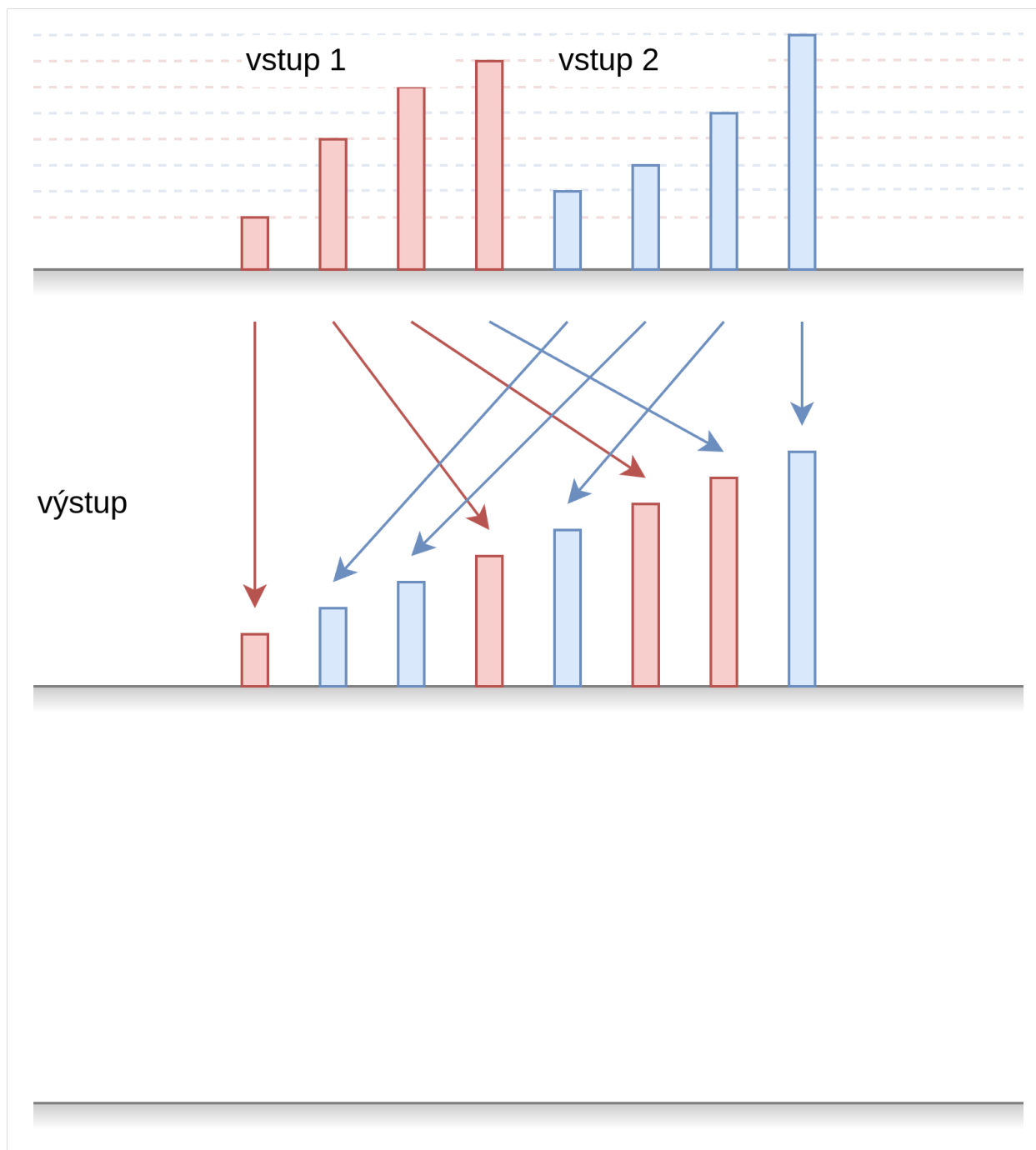
Merge sort

Vladislav Novák

Merge (princíp)



Merge (princíp)



Merge - úvod



Vstupné
usporiadané
postupnosti

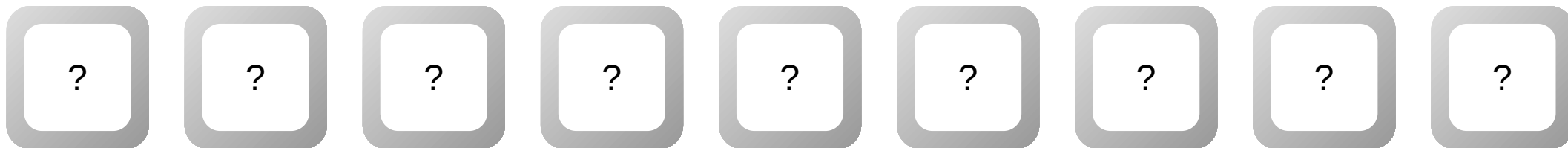
Merge - úvod



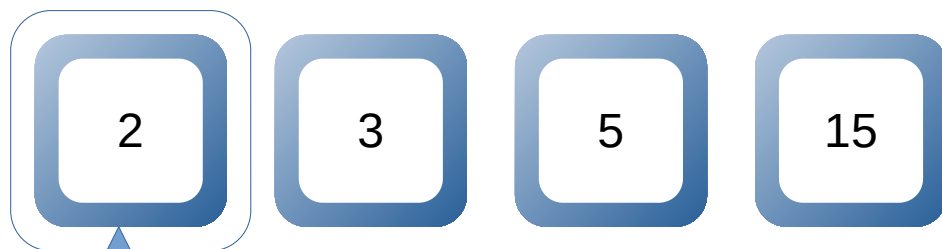
Vstupné
usporiadané
postupnosti



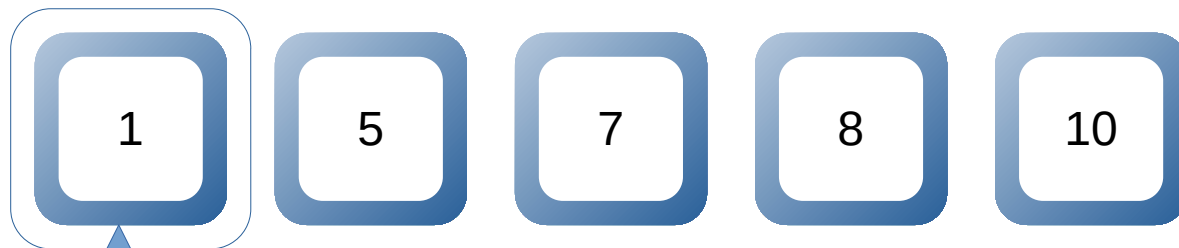
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



Merge - úvod



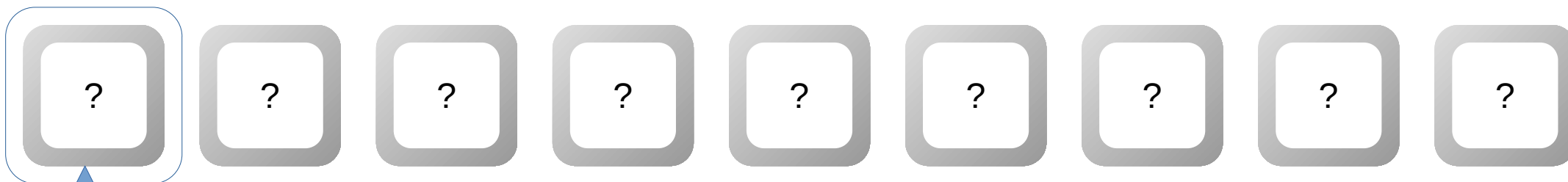
in1



in2

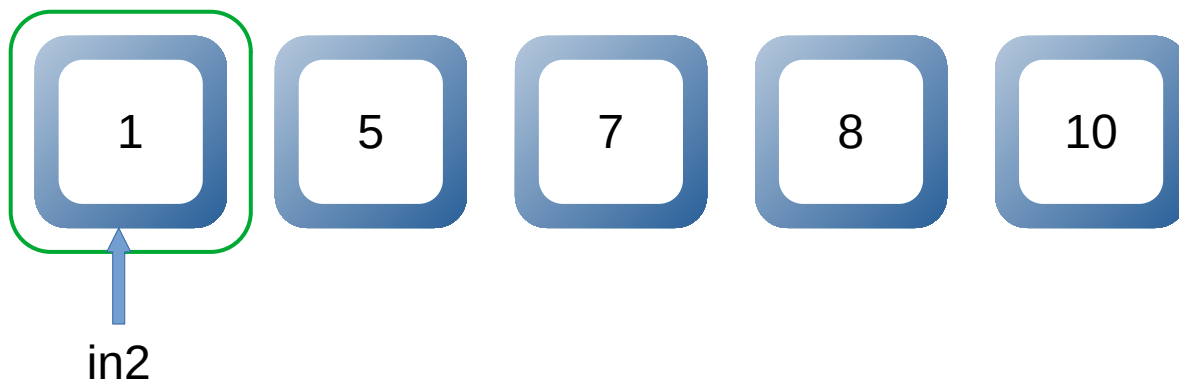
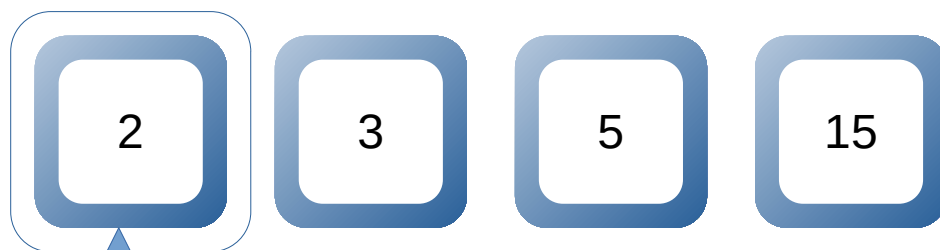
Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



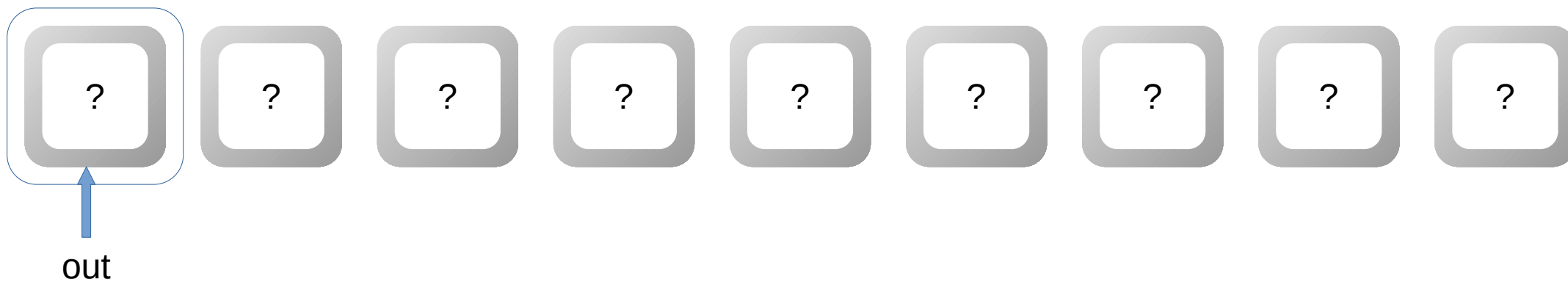
out

Merge - úvod

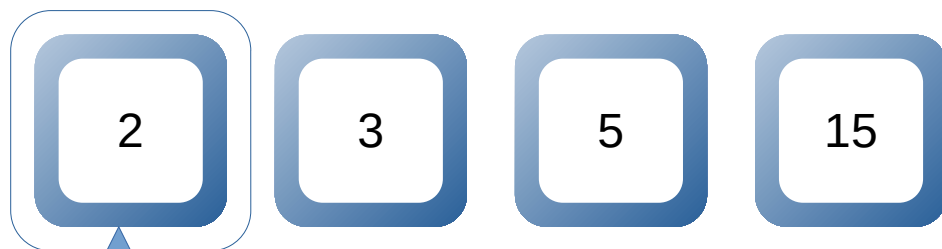


Vstupné
usporiadané
postupnosti

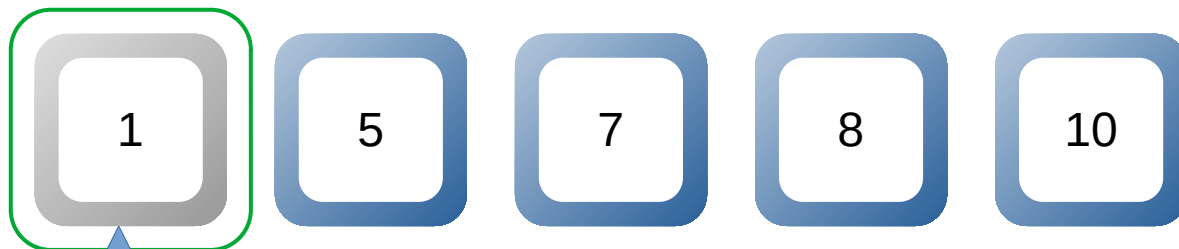
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



Merge - úvod



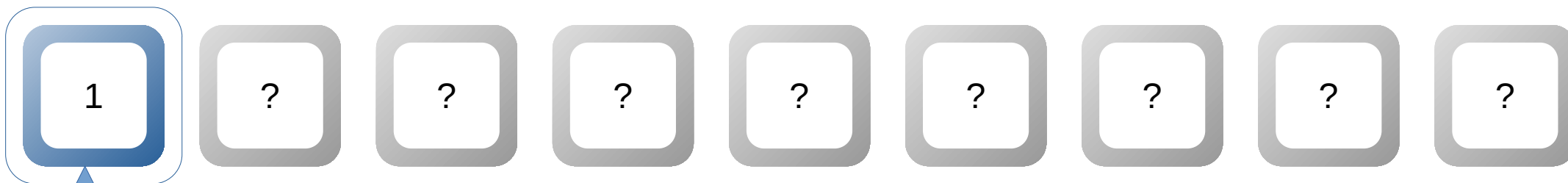
in1



in2

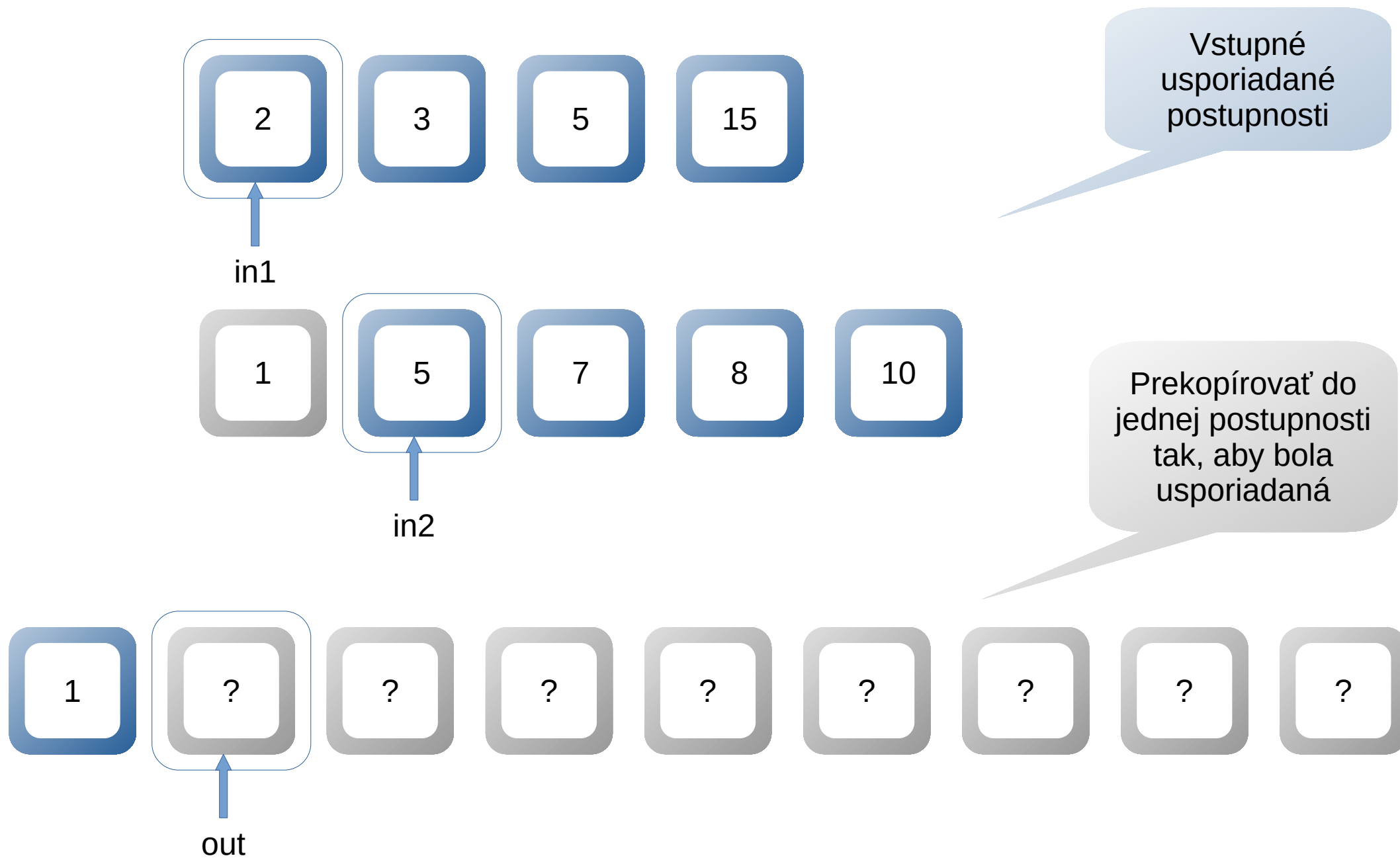
Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná

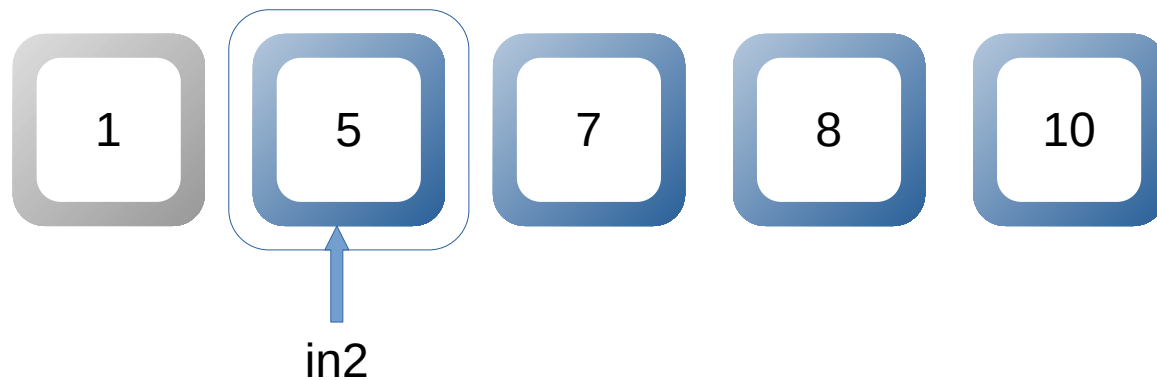
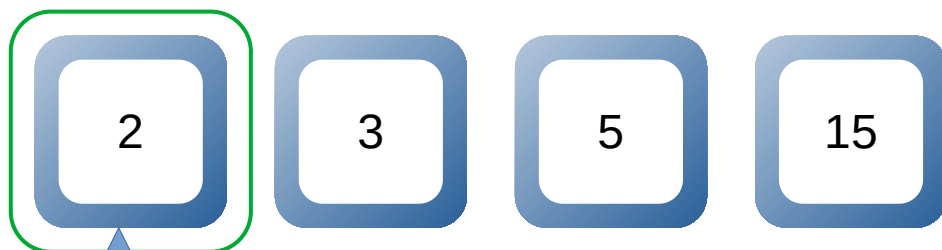


out

Merge - úvod

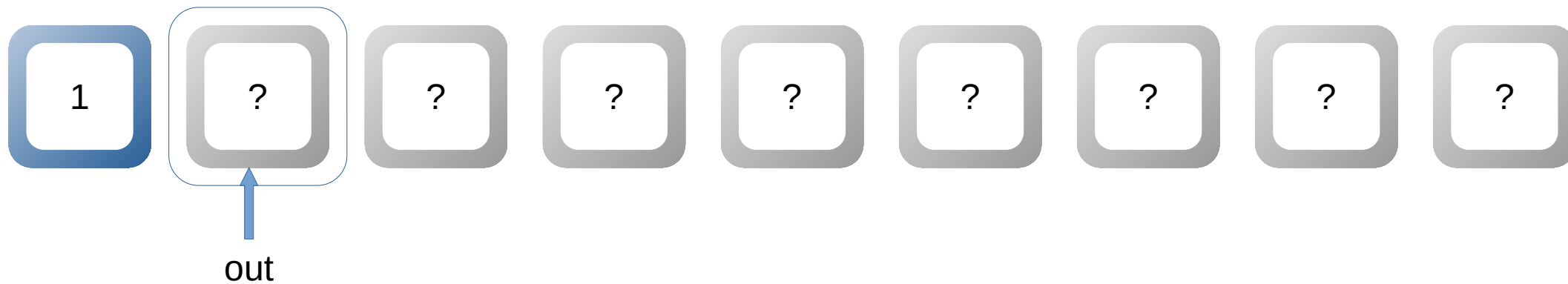


Merge - úvod

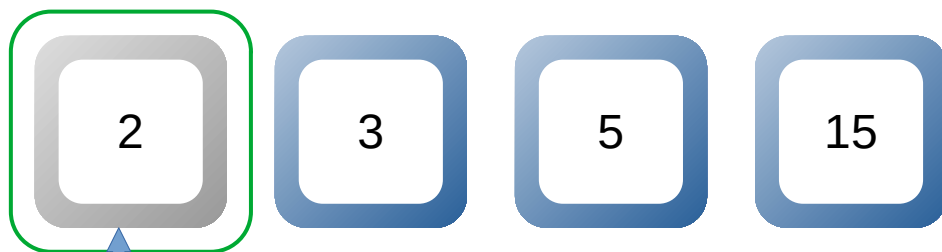


Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



Merge - úvod



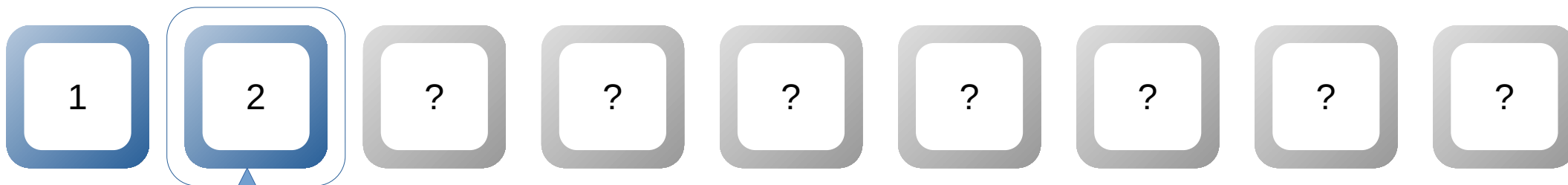
in1



in2

Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



out

Merge - úvod



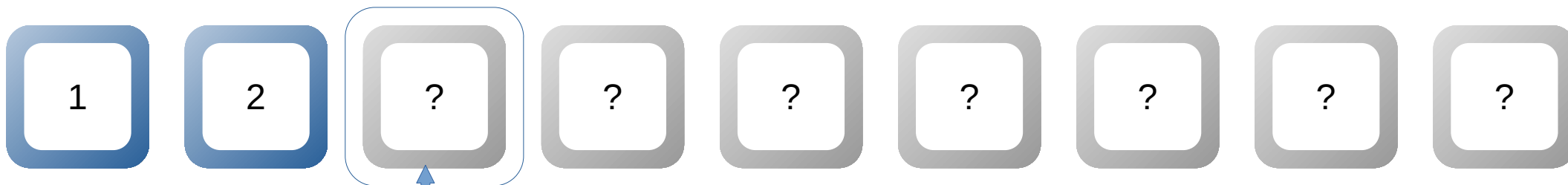
in1



in2

Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná

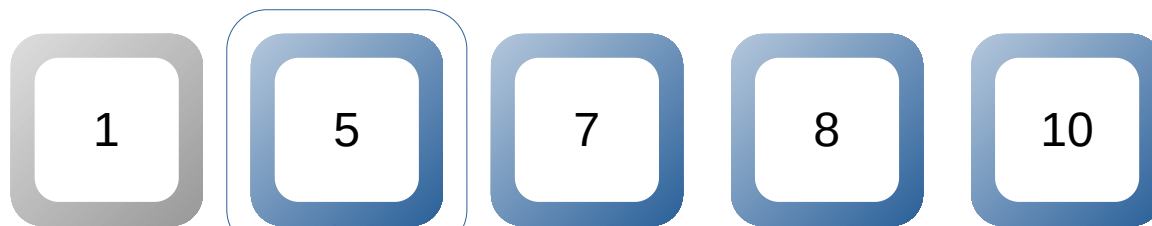


out

Merge - úvod



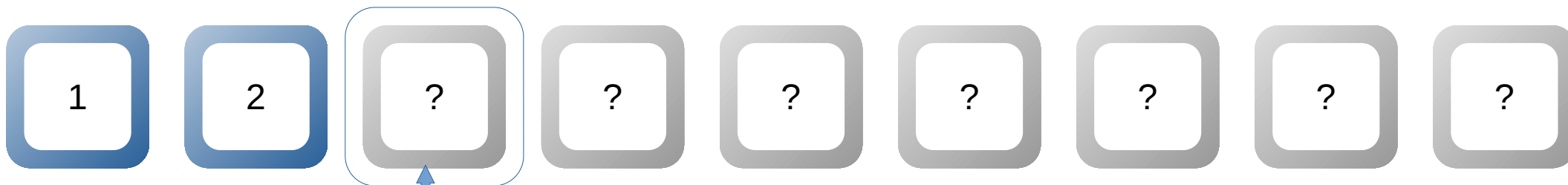
in1



in2

Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



out

Merge - úvod



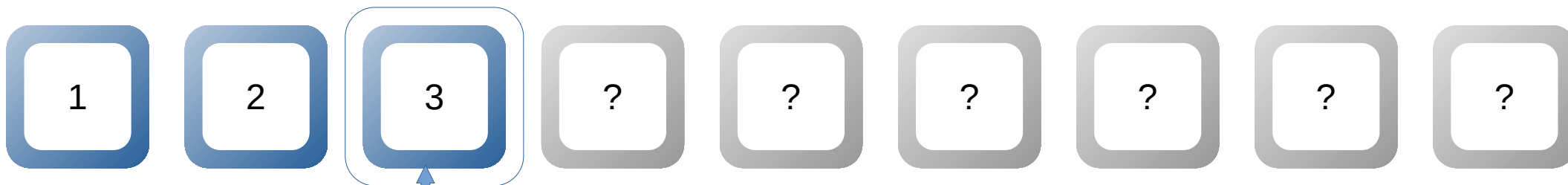
in1



in2

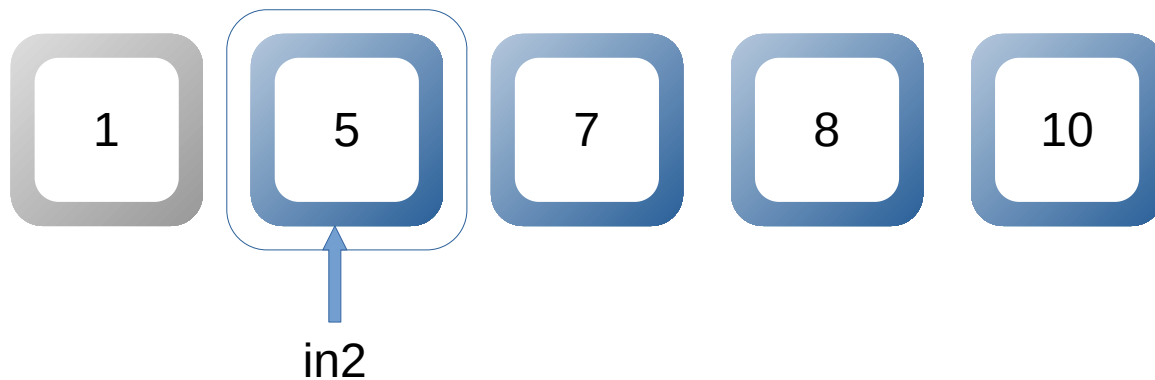
Vstupné
usporiadané
postupnosti

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



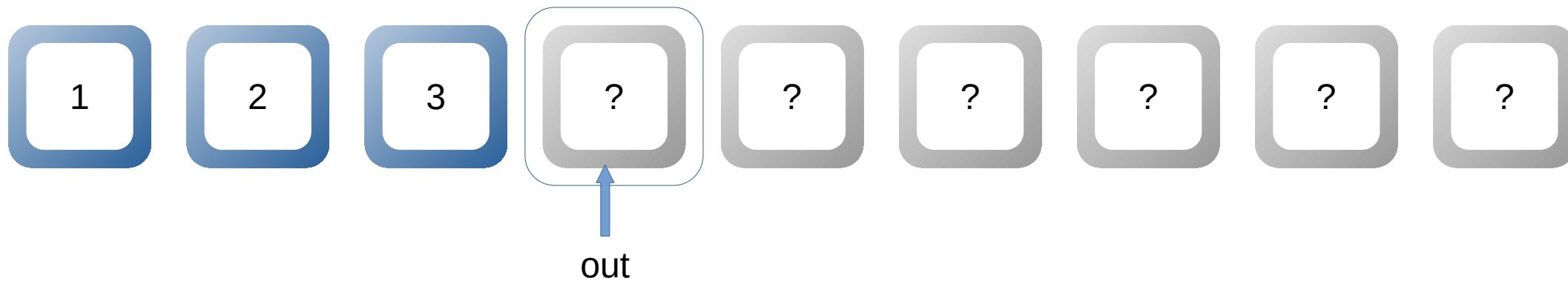
out

Merge - úvod



Vstupné
usporiadané
postupnosti

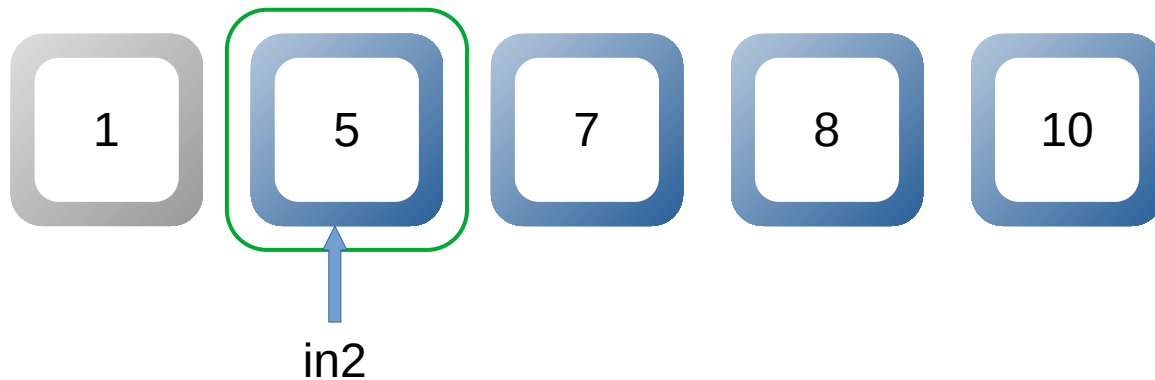
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



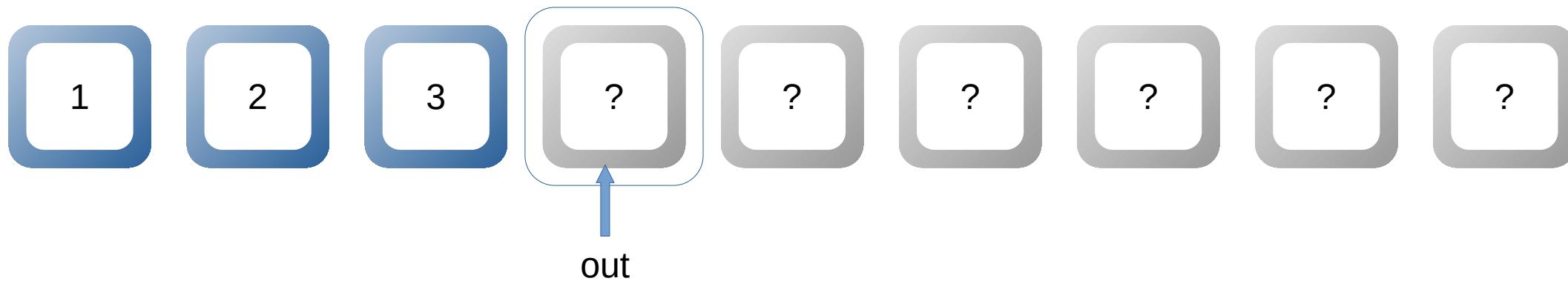
Merge - úvod



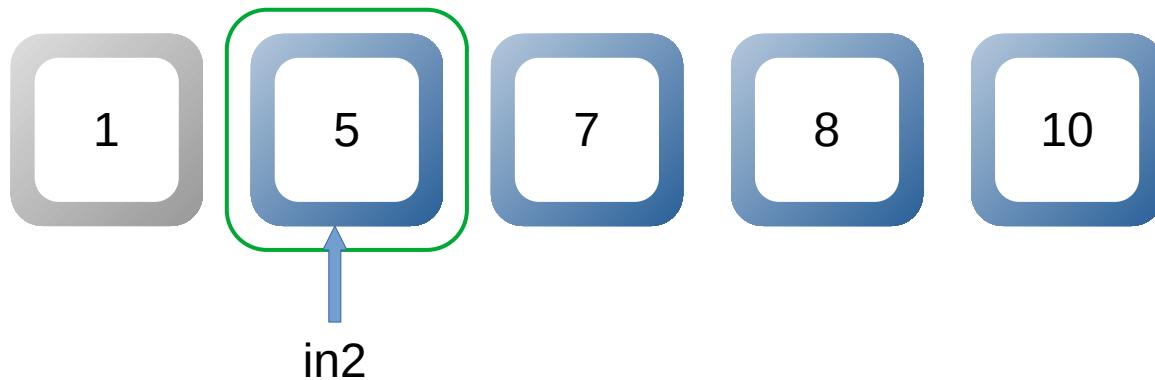
Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná

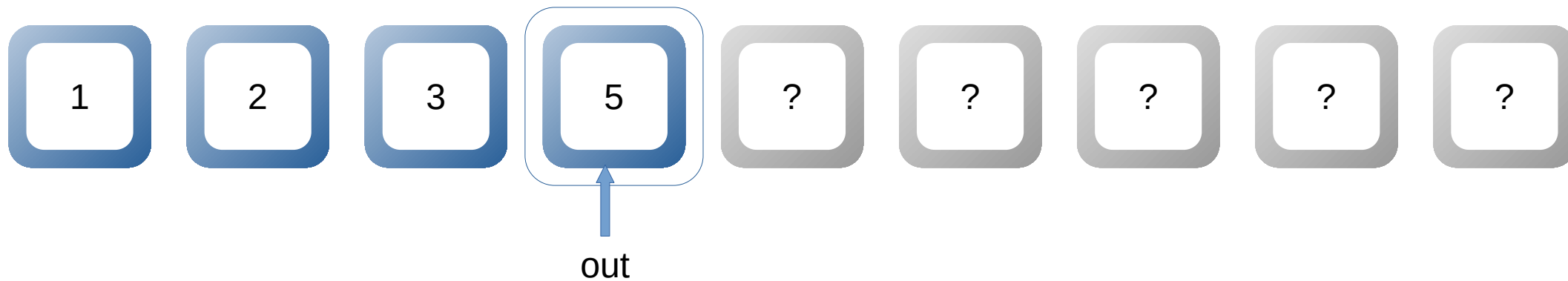


Merge - úvod



Vstupné
usporiadané
postupnosti

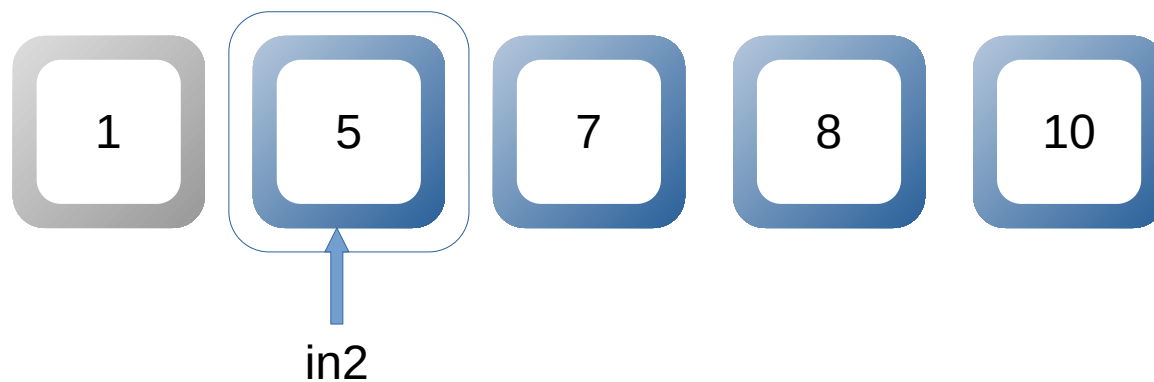
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



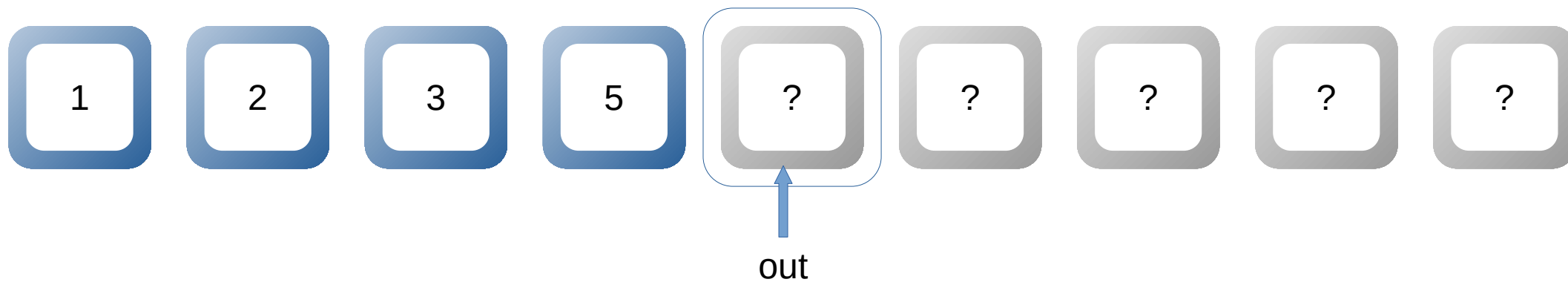
Merge - úvod



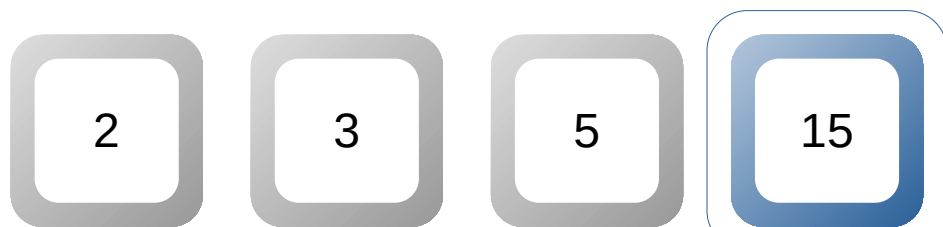
Vstupné
usporiadané
postupnosti



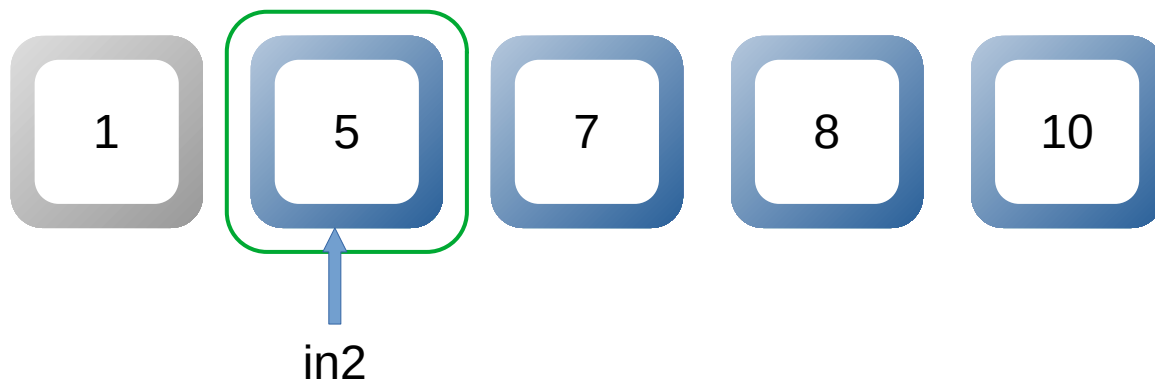
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



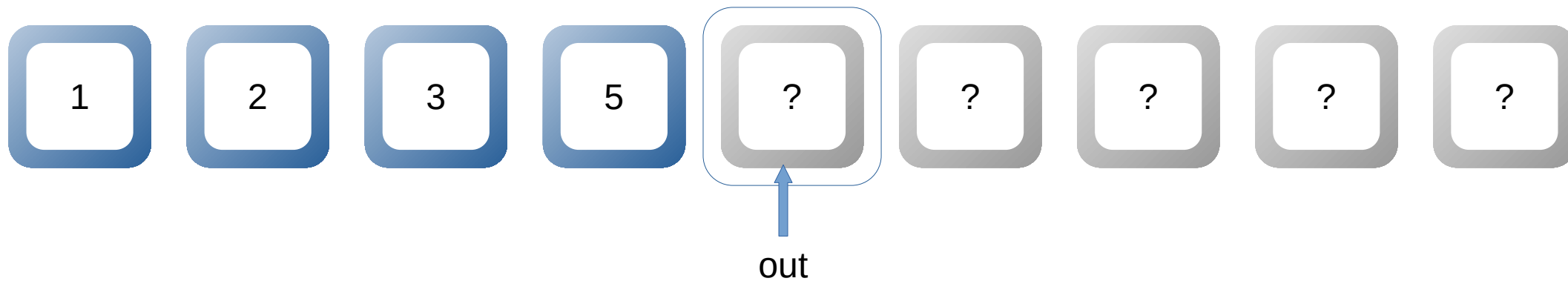
Merge - úvod



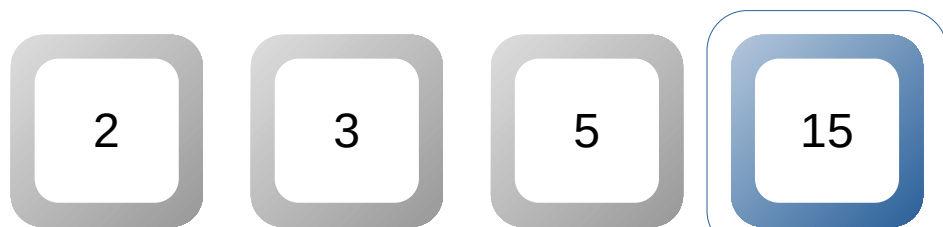
Vstupné
usporiadané
postupnosti



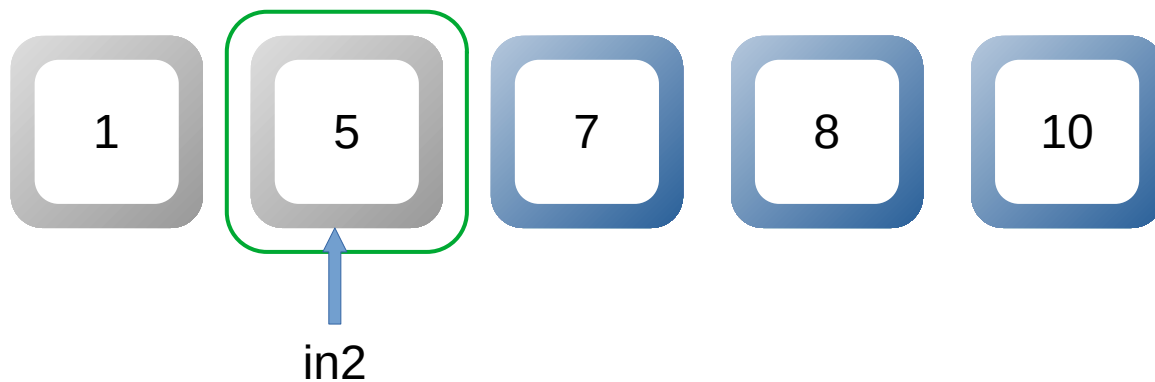
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



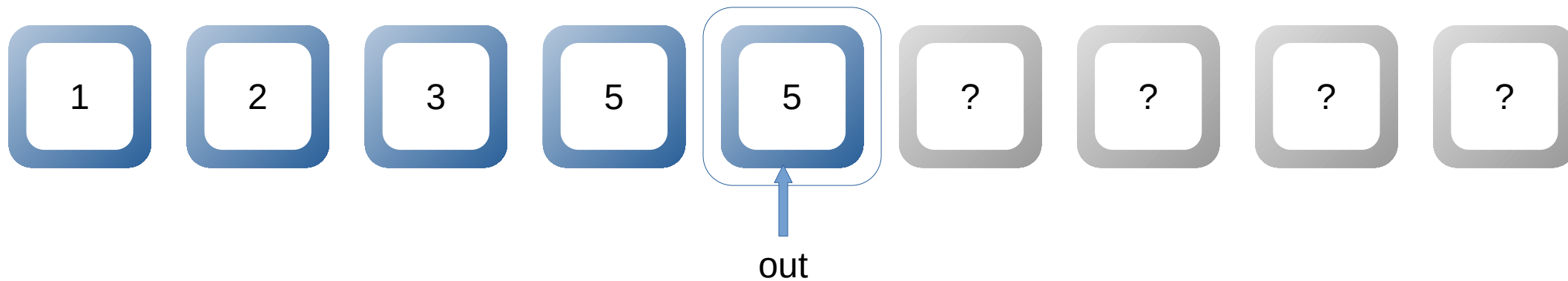
Merge - úvod



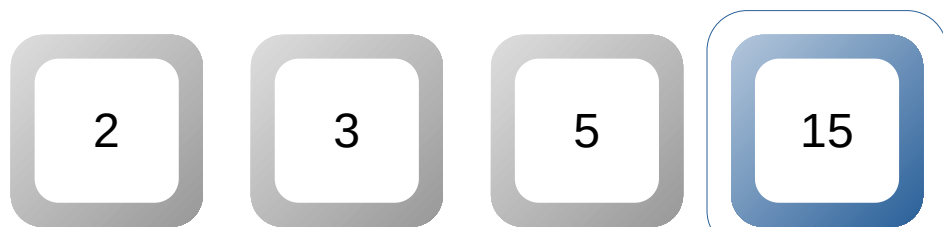
Vstupné
usporiadané
postupnosti



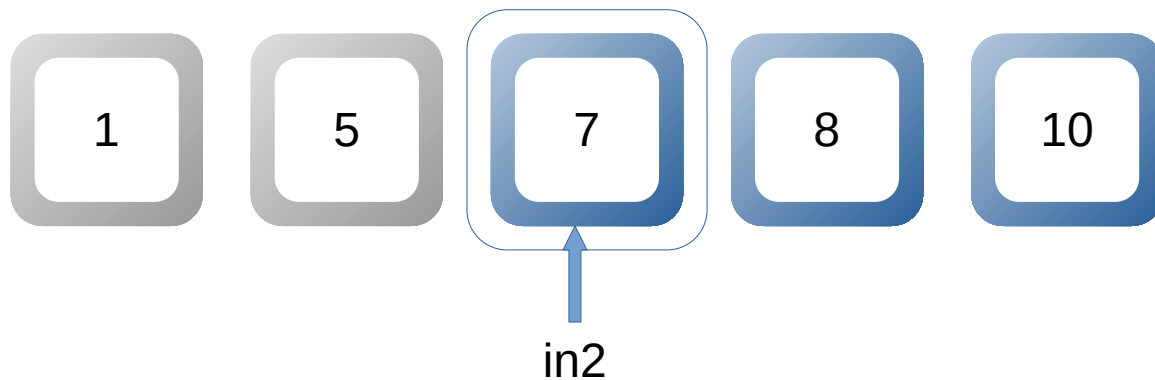
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



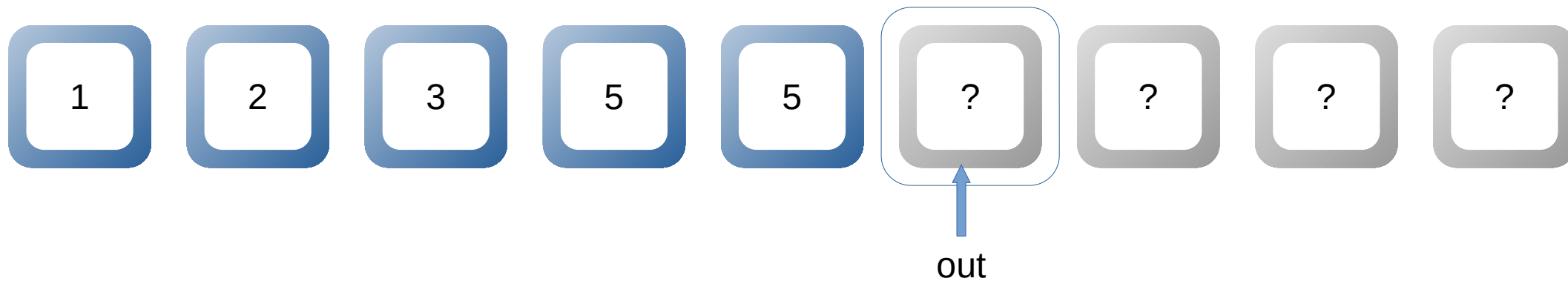
Merge - úvod



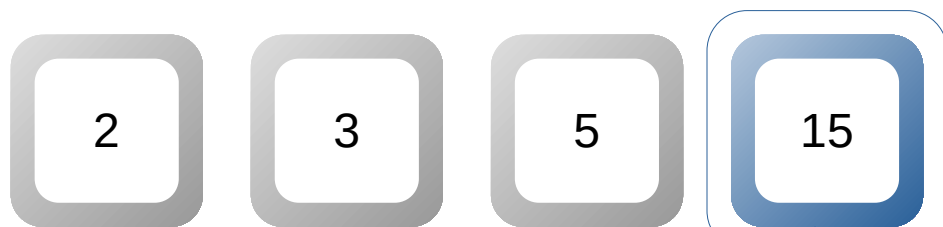
Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



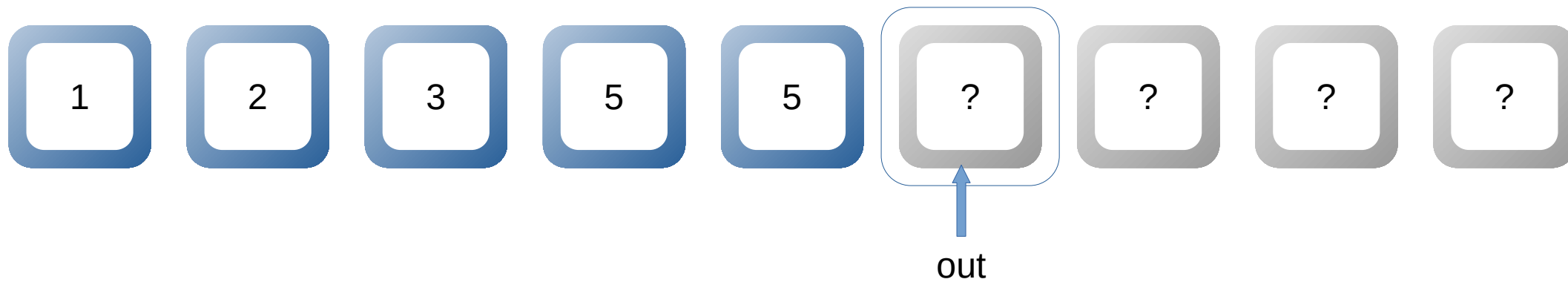
Merge - úvod



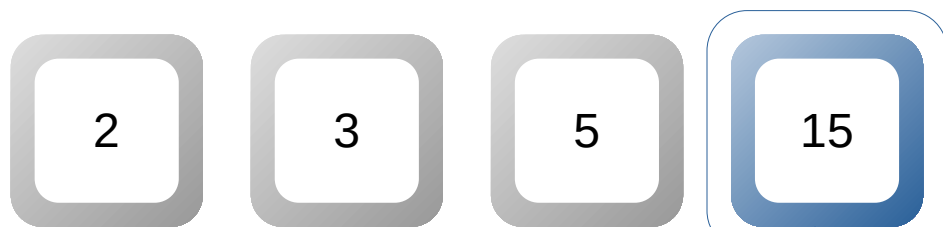
Vstupné
usporiadané
postupnosti



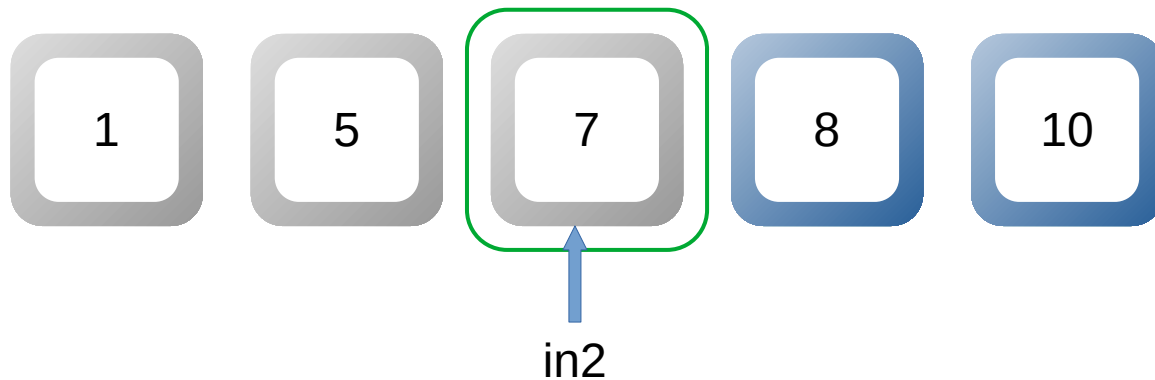
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



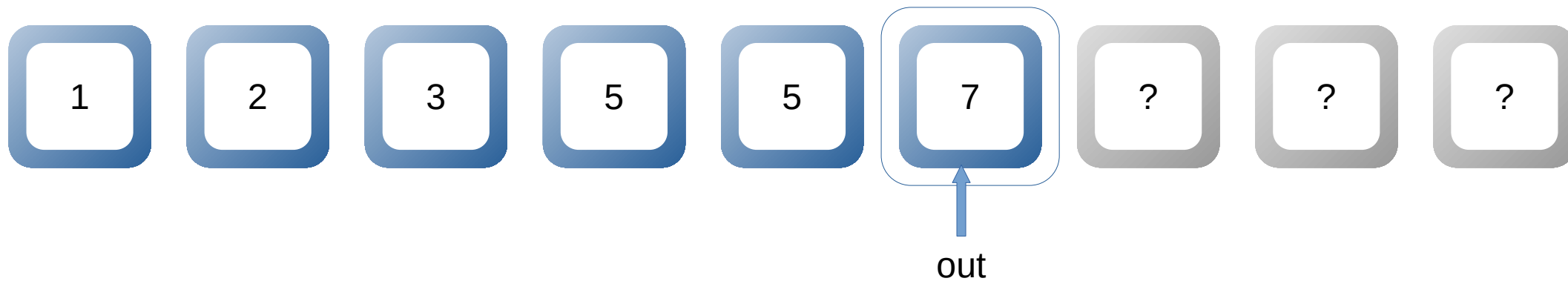
Merge - úvod



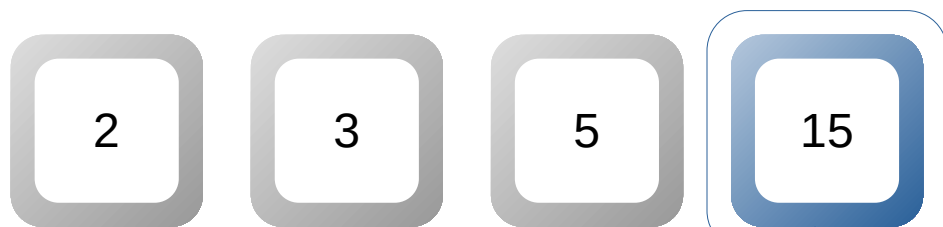
Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



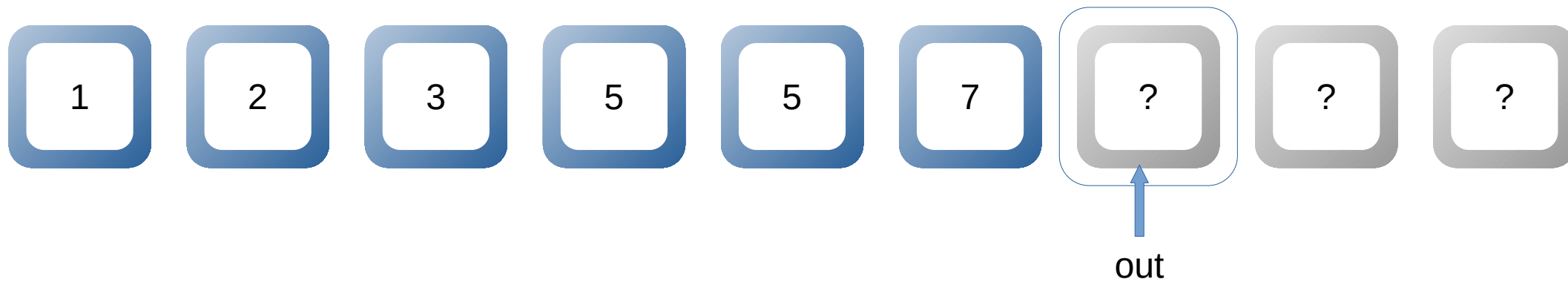
Merge - úvod



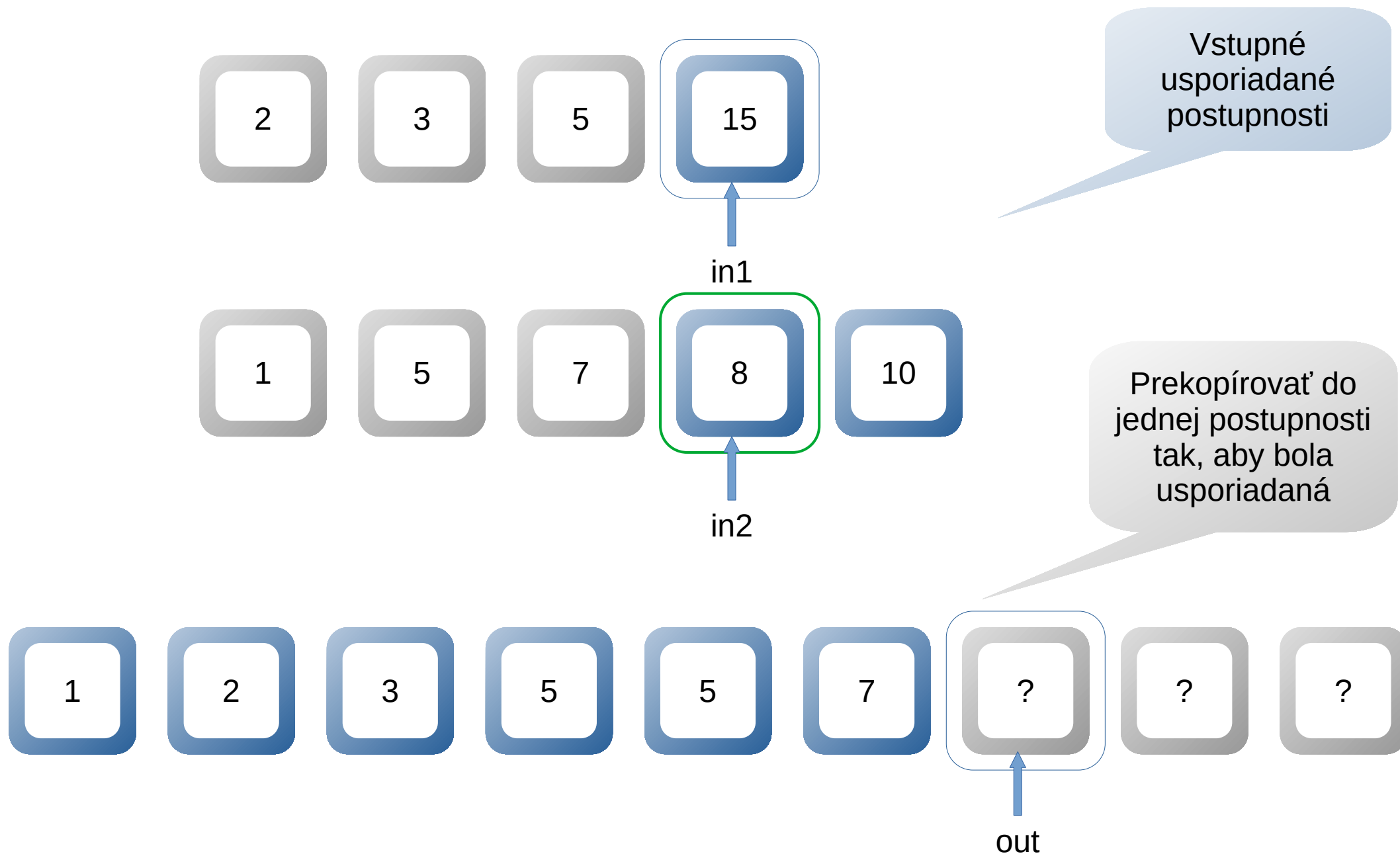
Vstupné
usporiadané
postupnosti



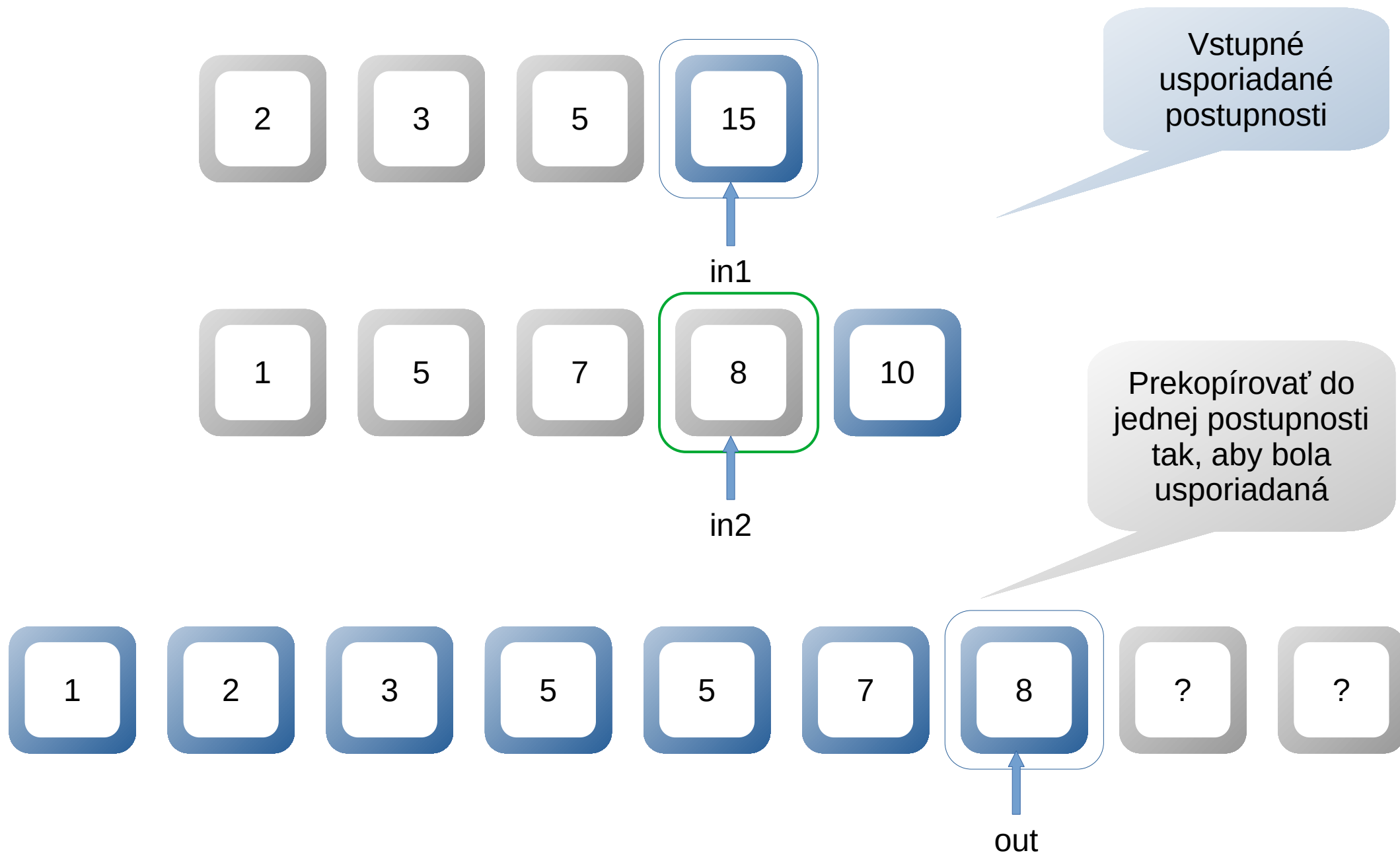
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



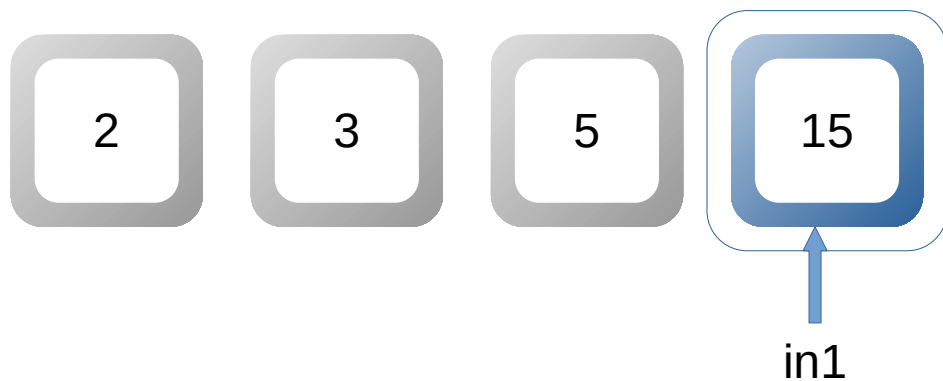
Merge - úvod



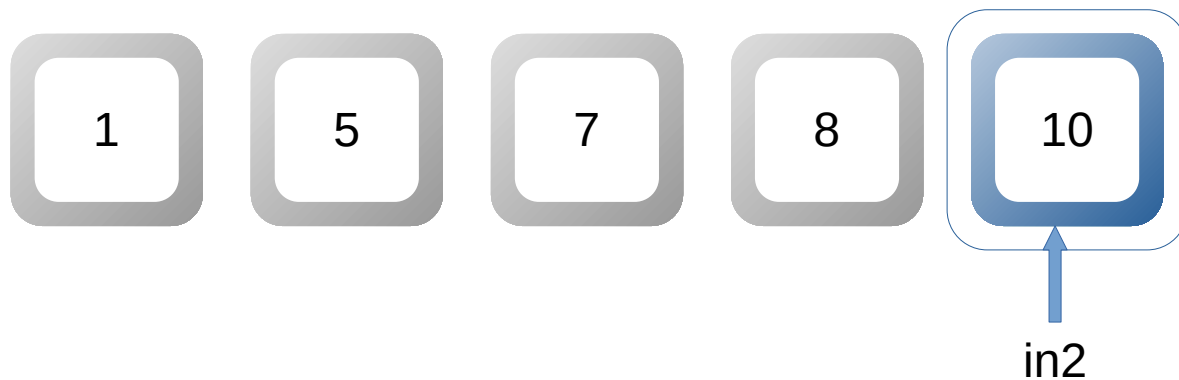
Merge - úvod



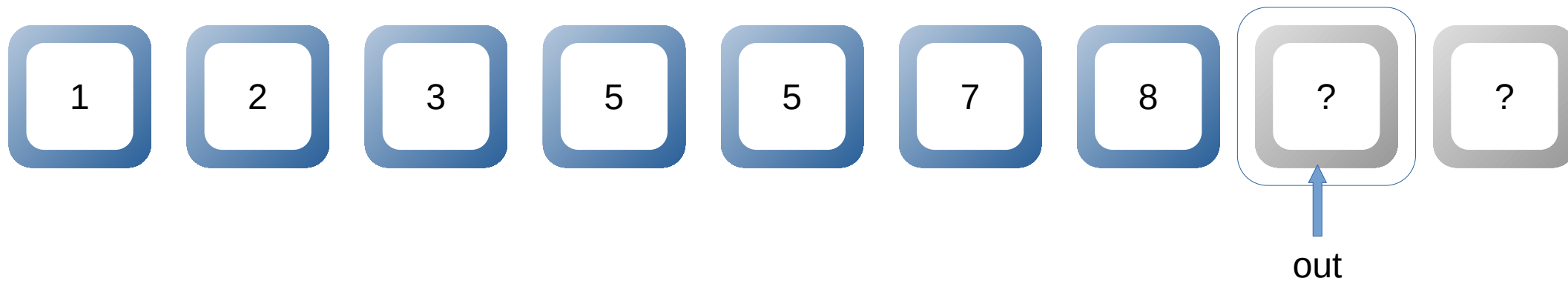
Merge - úvod



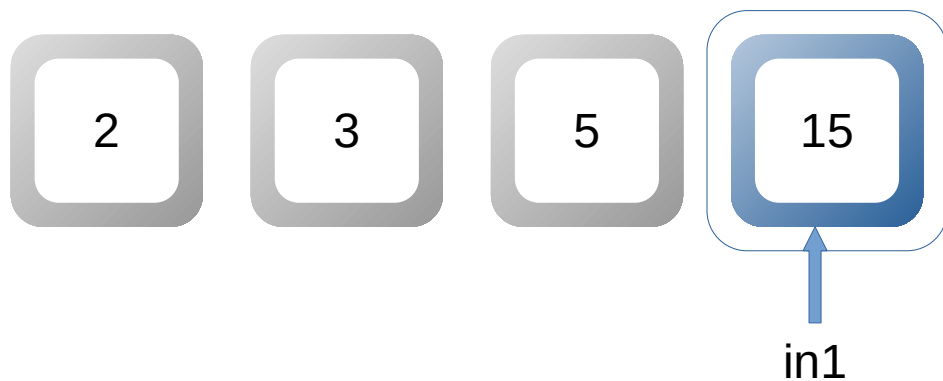
Vstupné
usporiadané
postupnosti



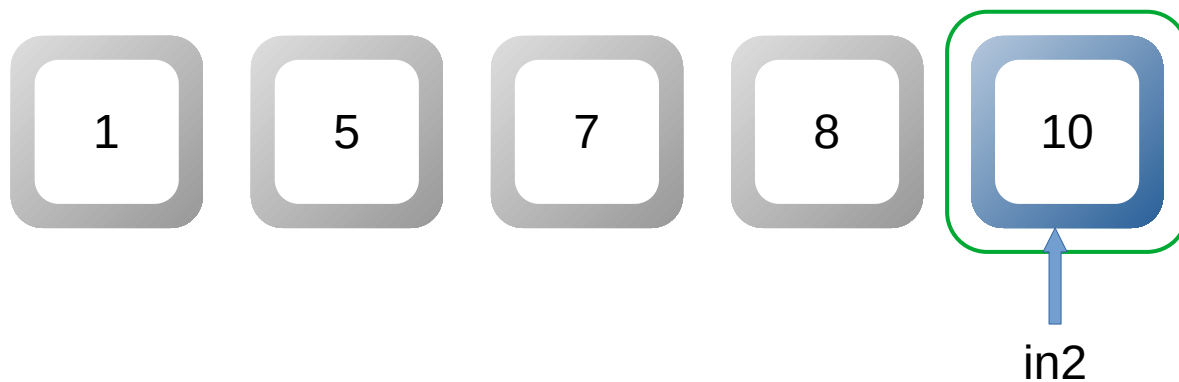
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



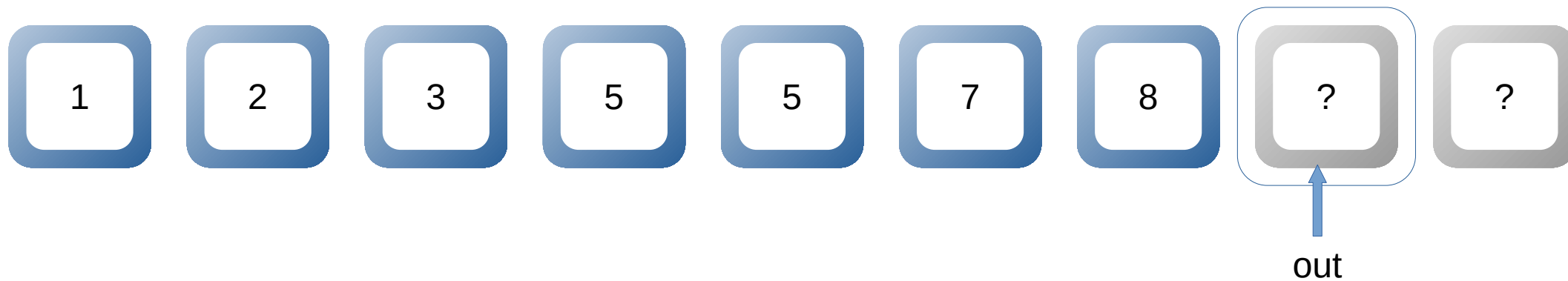
Merge - úvod



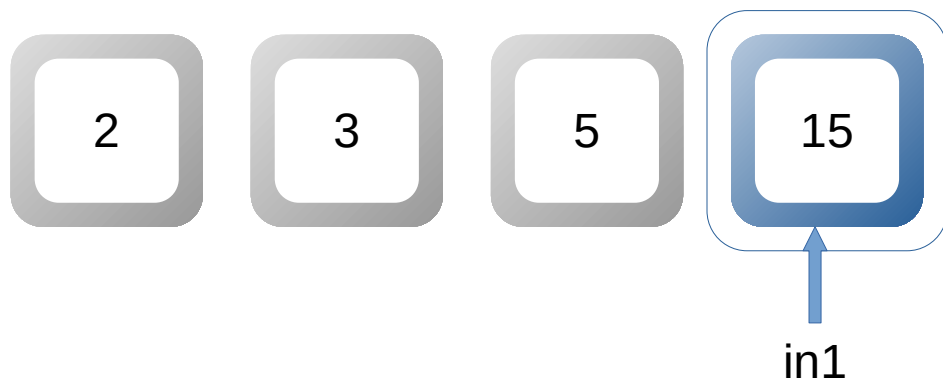
Vstupné
usporiadané
postupnosti



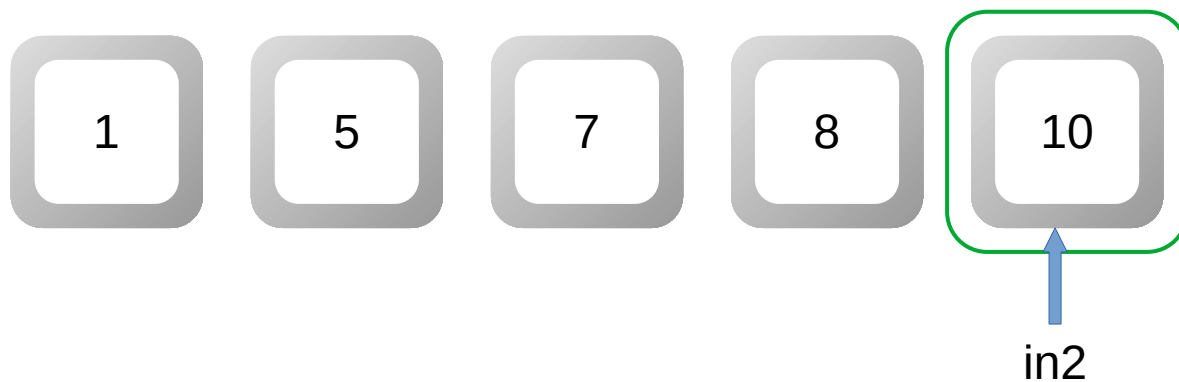
Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



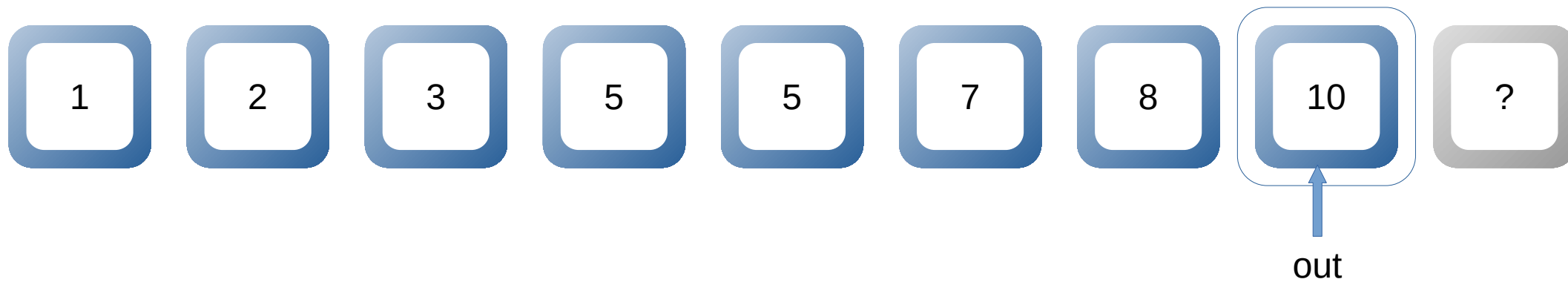
Merge - úvod



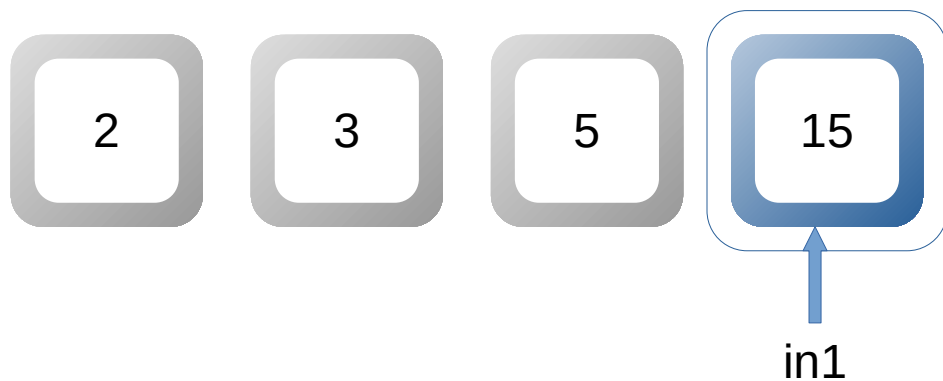
Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



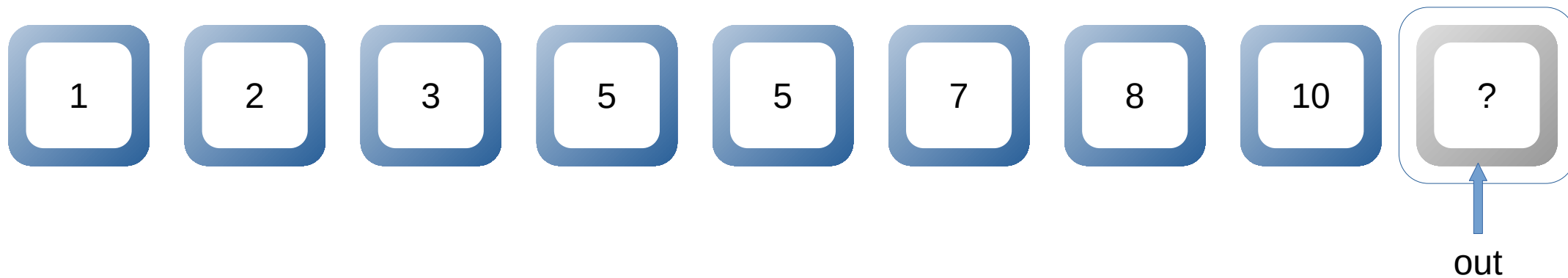
Merge - úvod



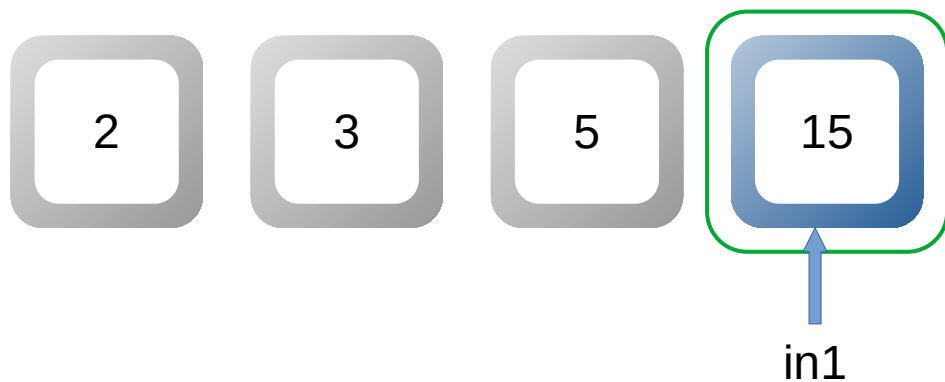
Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



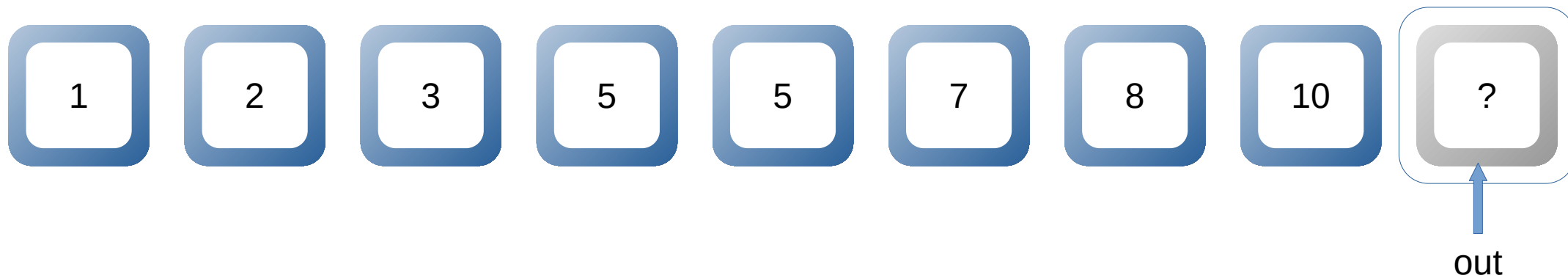
Merge - úvod



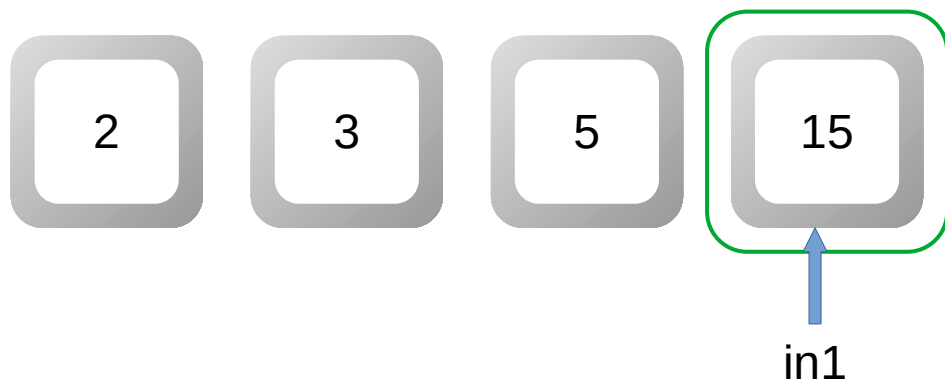
Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



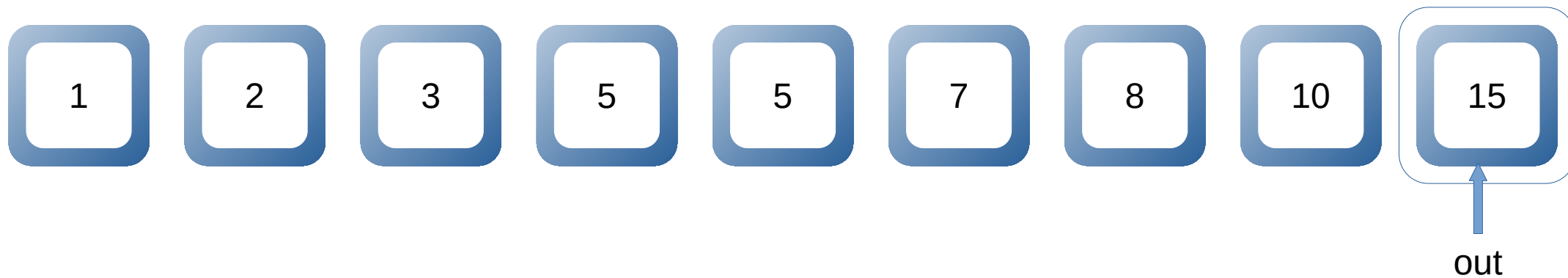
Merge - úvod



Vstupné
usporiadané
postupnosti



Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná



Merge - úvod



↑
in1

Vstupné
usporiadané
postupnosti



↑
in2

Prekopírovať do
jednej postupnosti
tak, aby bola
usporiadaná

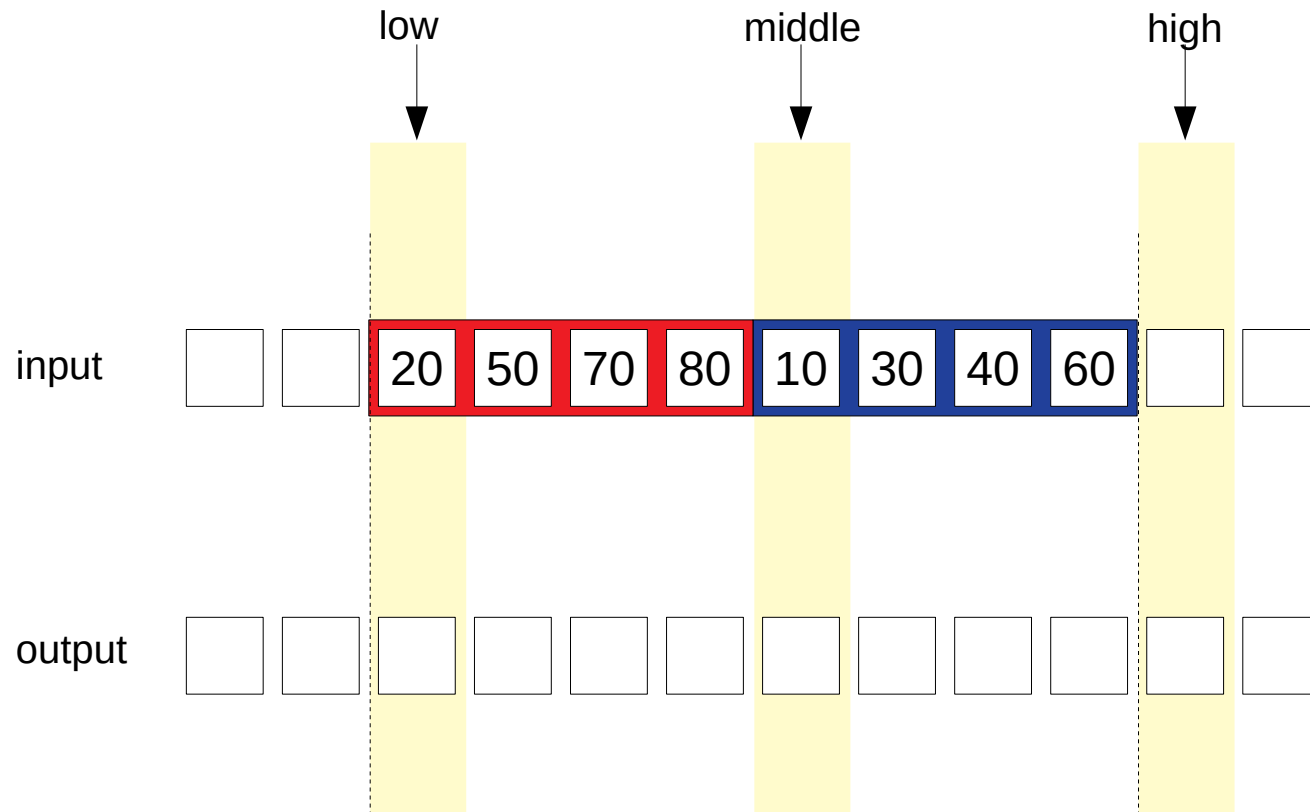


↑
ou

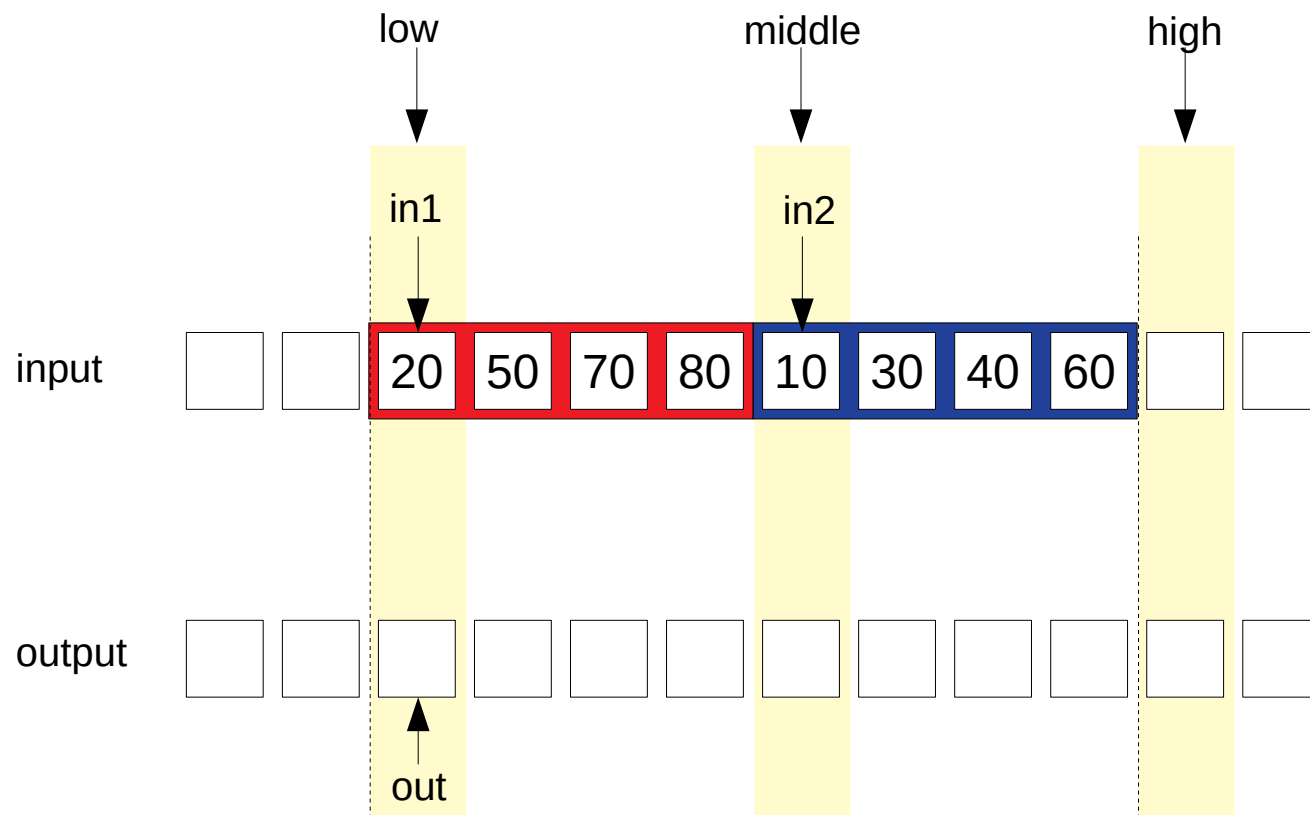
Merge - úvod

Začneme od jednoprvkových postupností. Jednoprvková postupnosť je usporiadaná

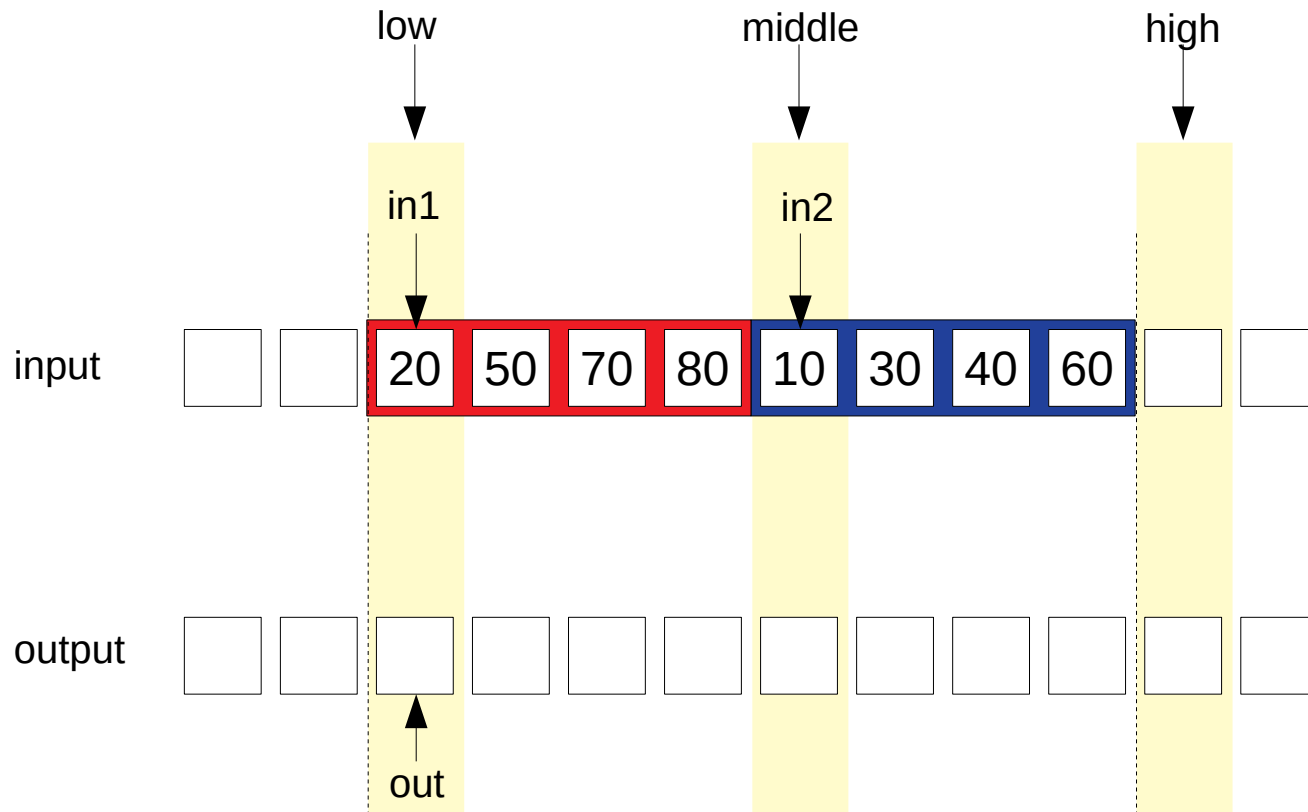
Merge susedných častí poľa



Merge susedných častí poľa

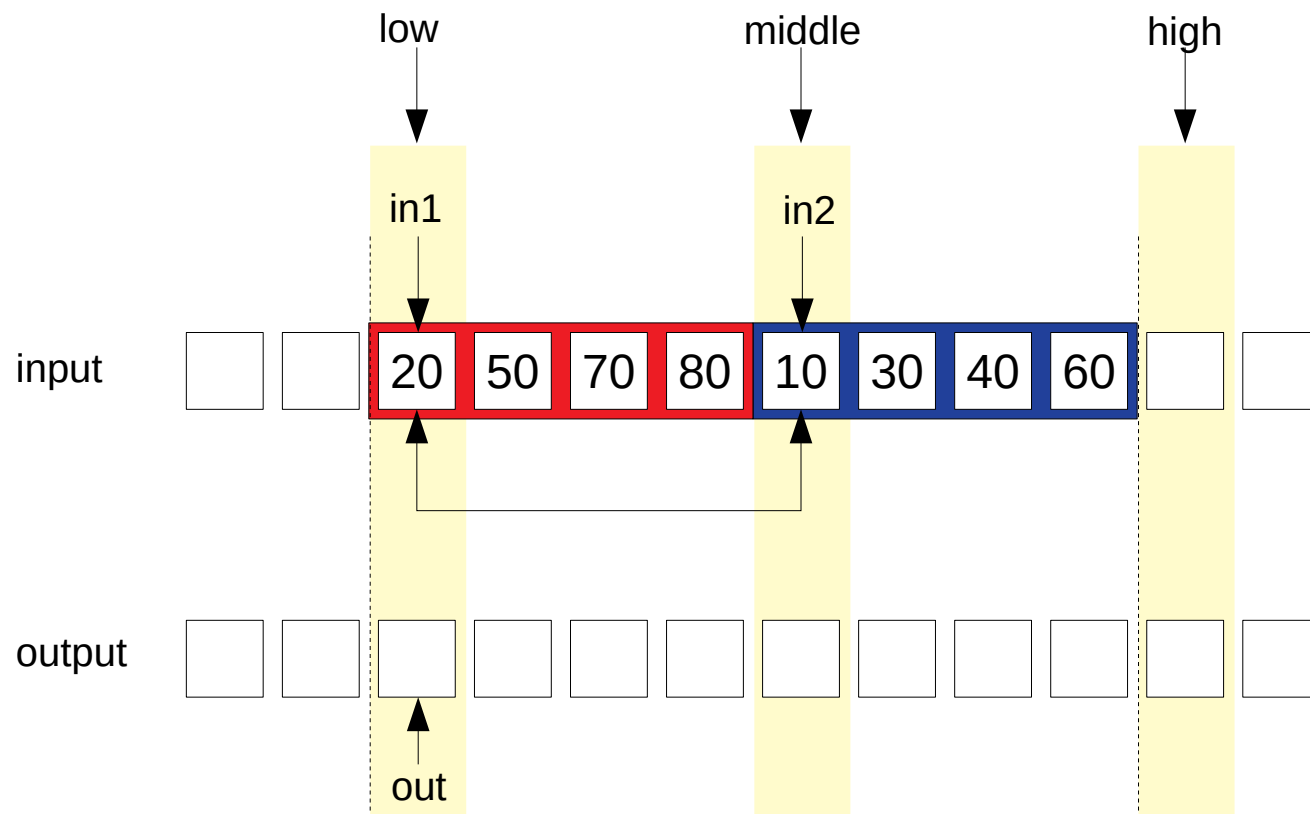


Merge susedných častí poľa



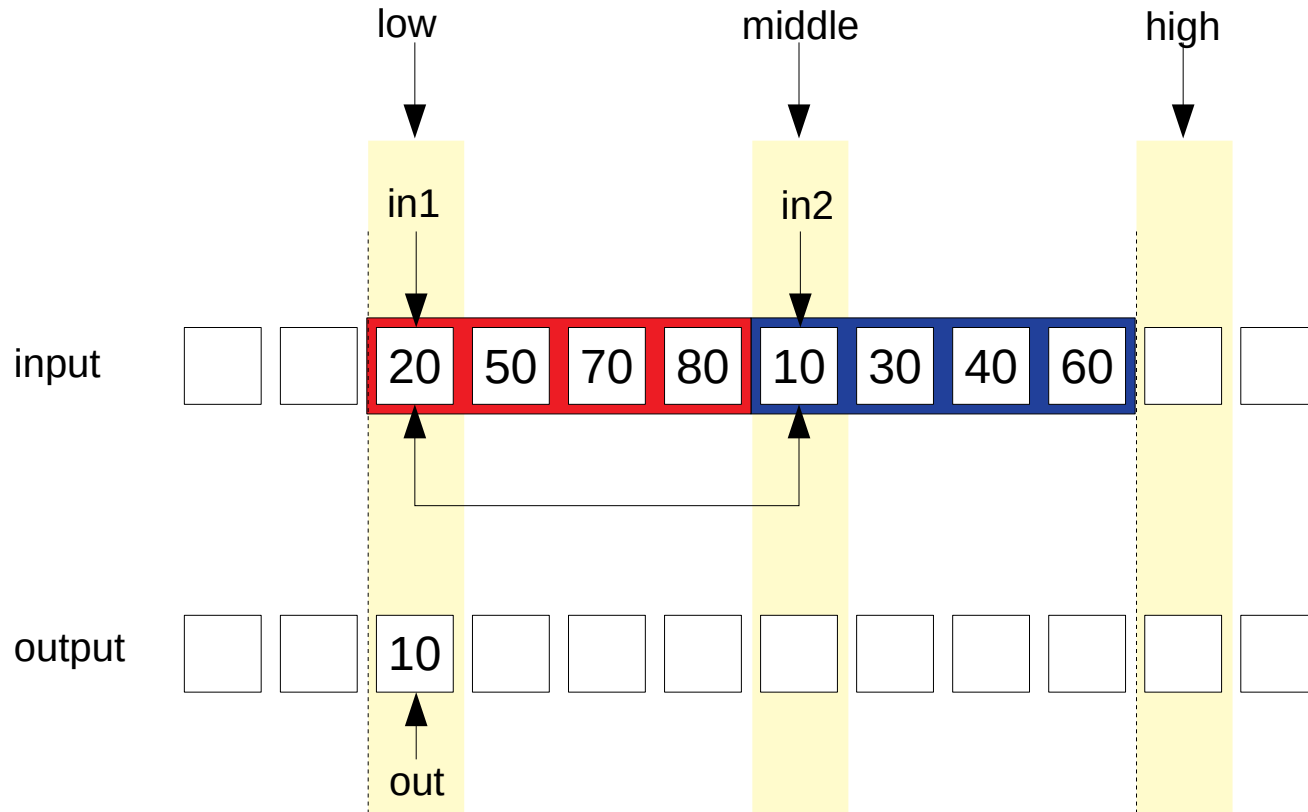
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



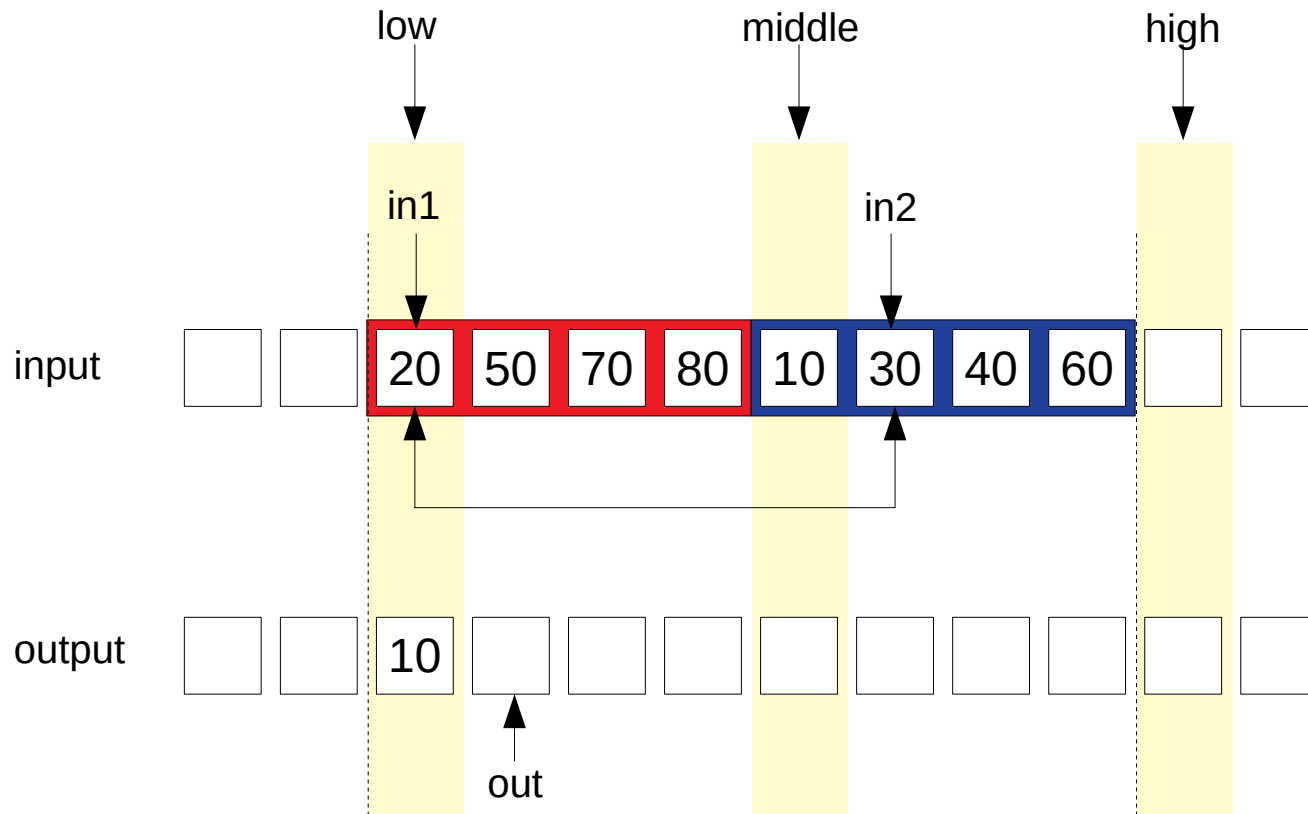
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



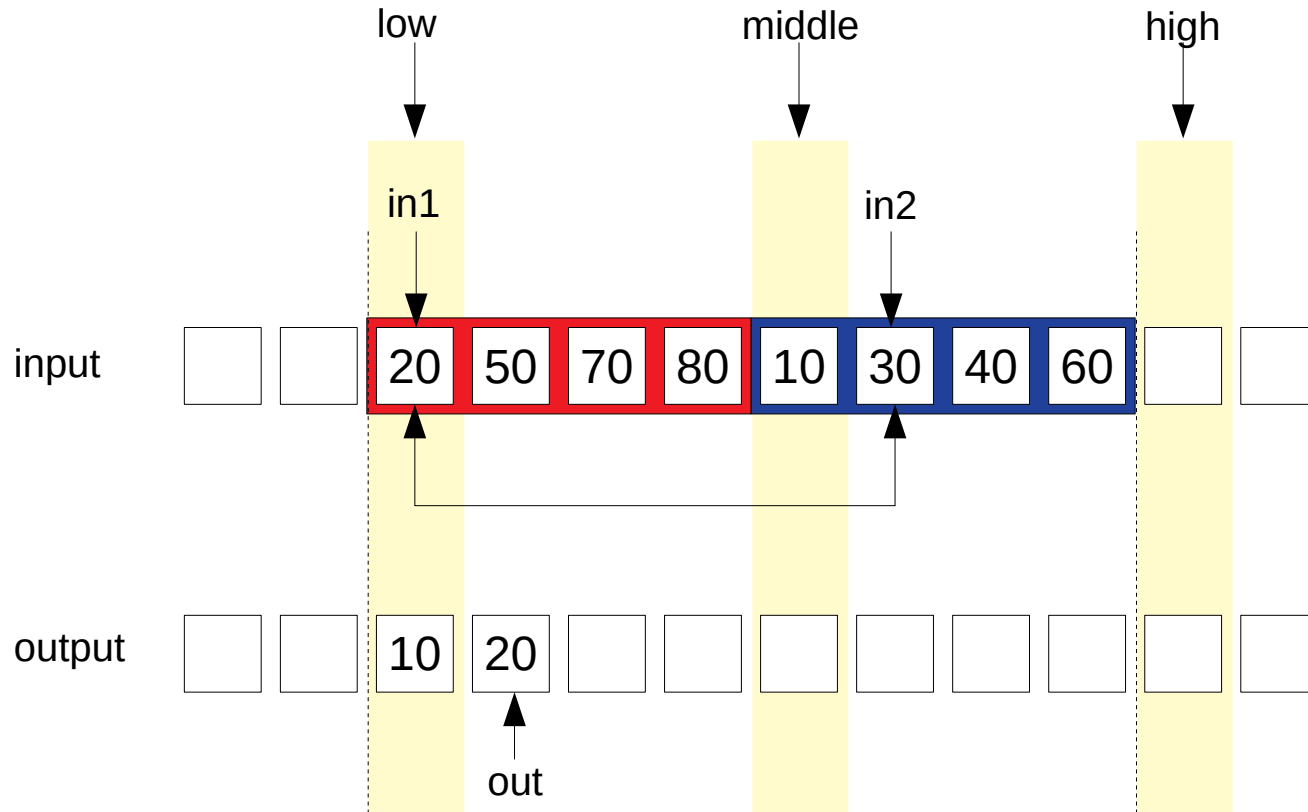
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```


Merge susedných častí poľa



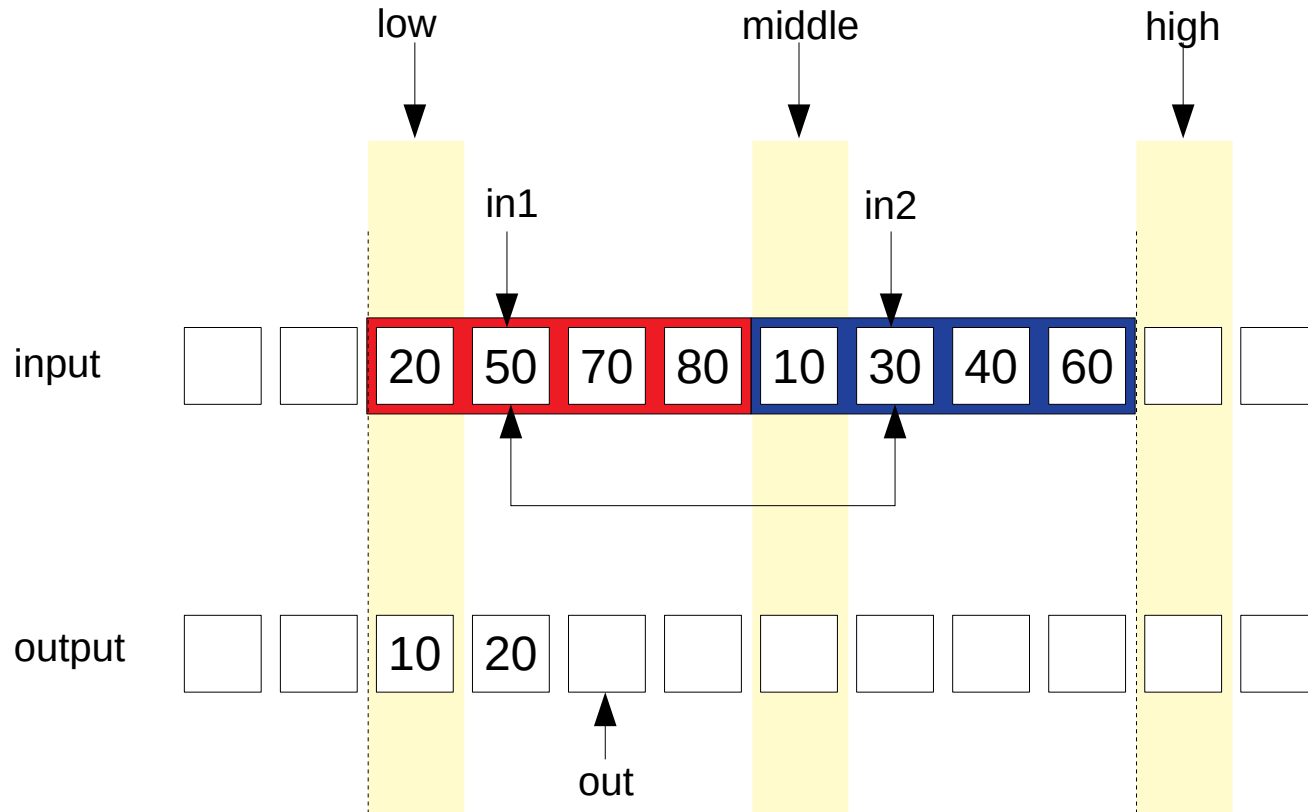
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



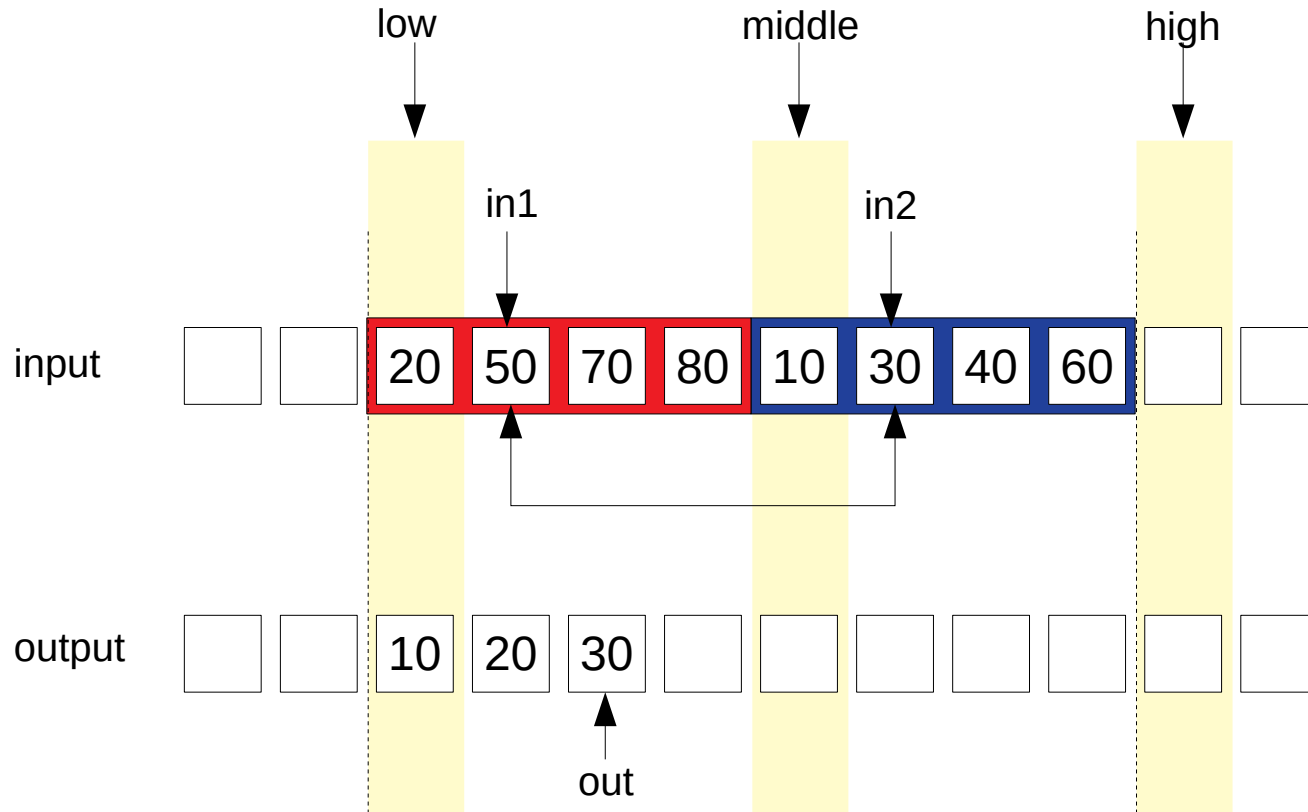
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



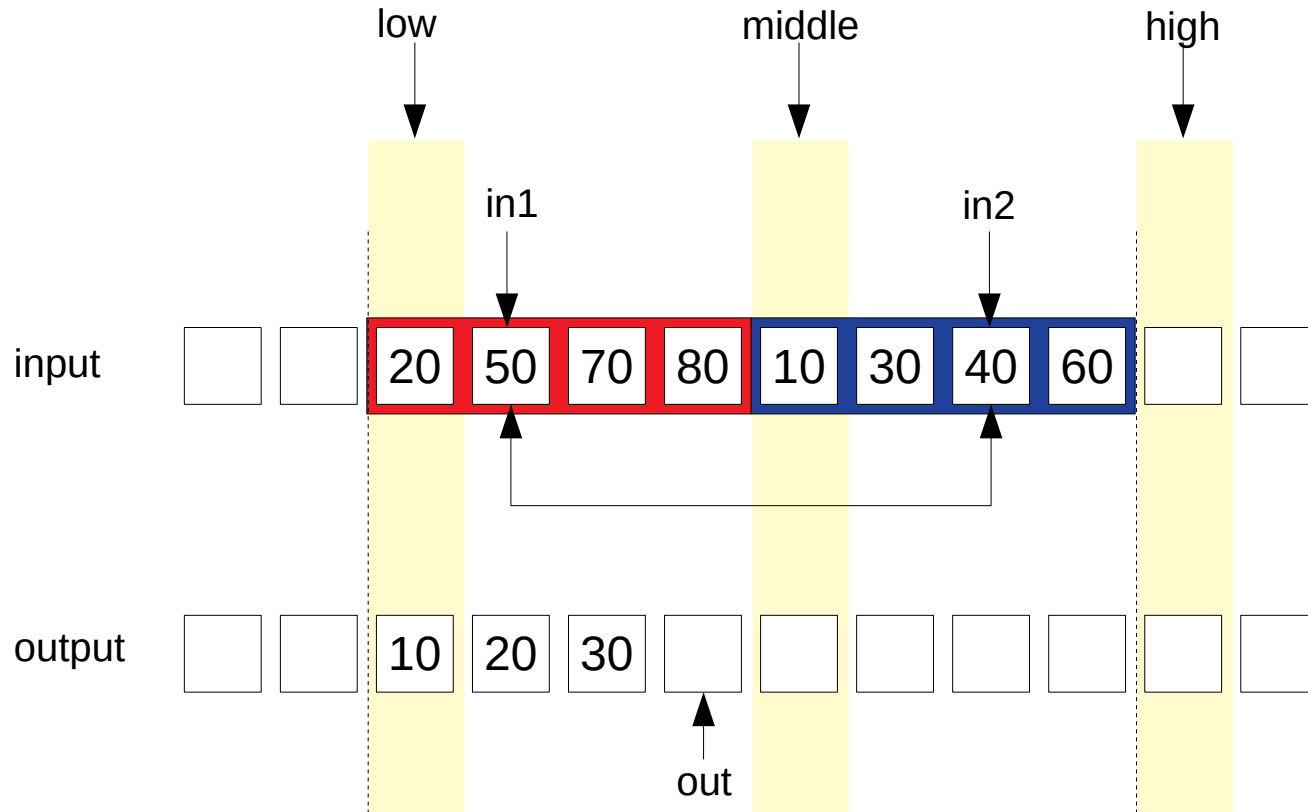
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



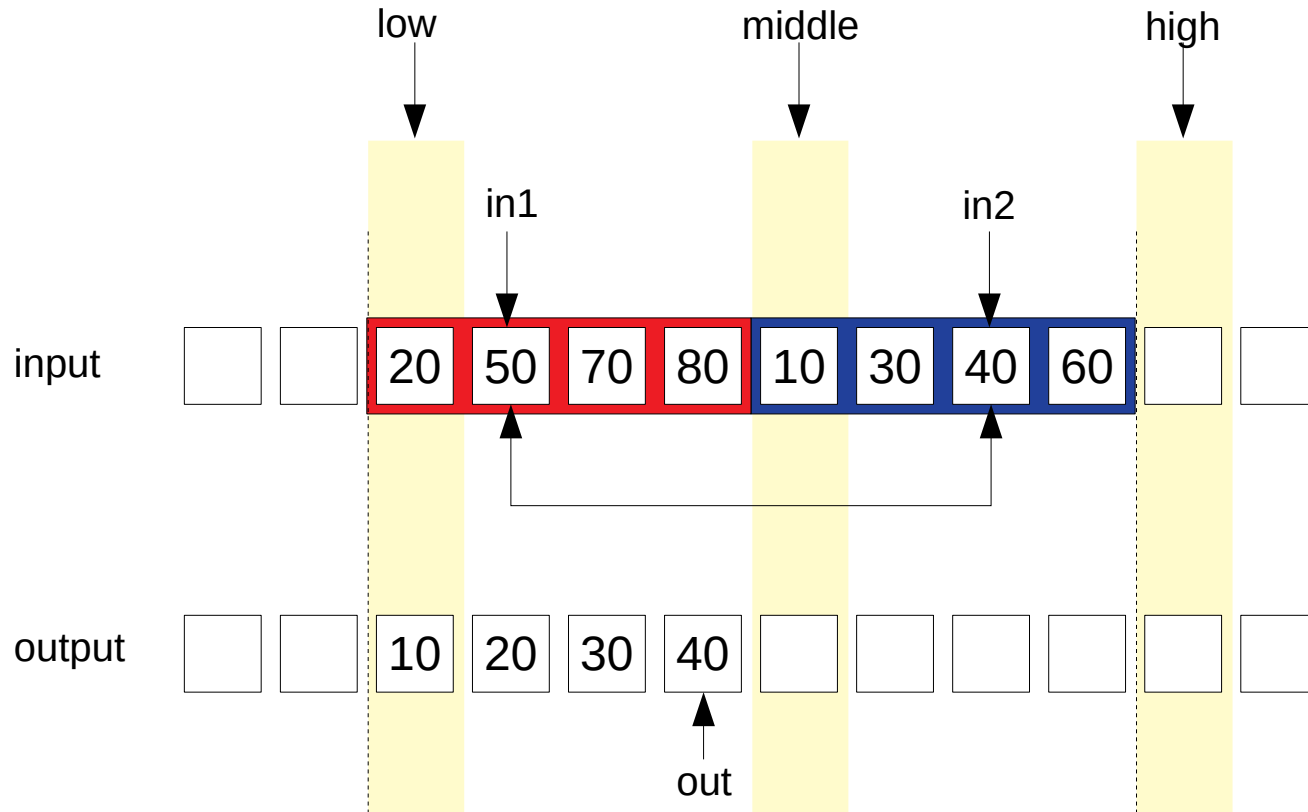
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



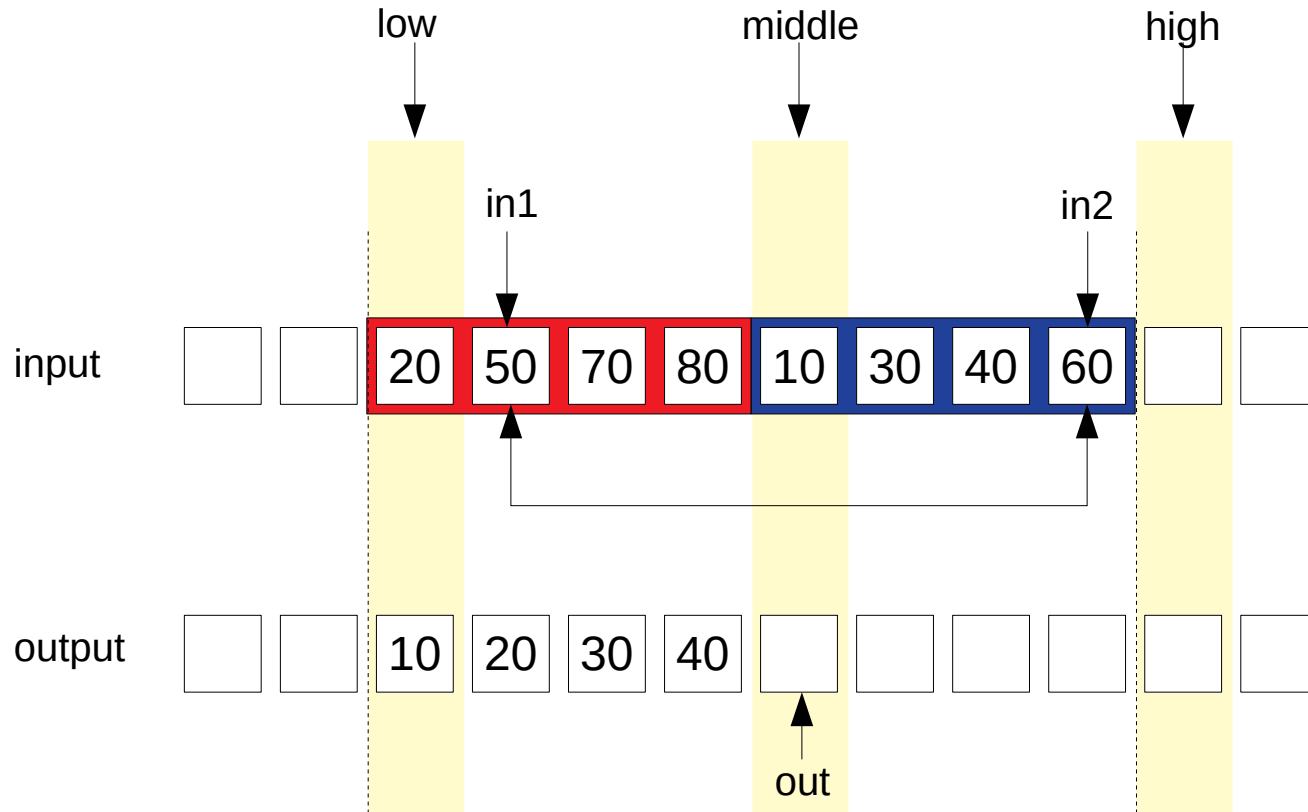
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



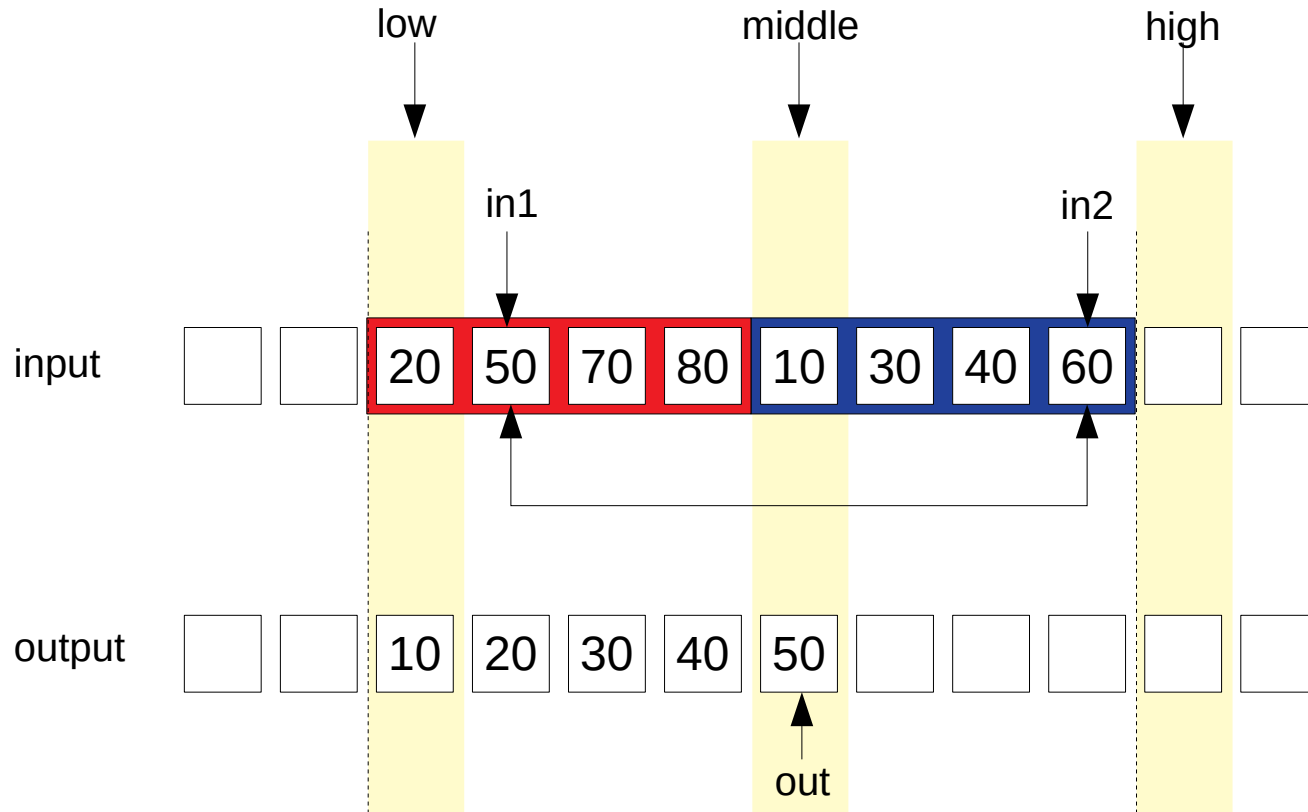
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



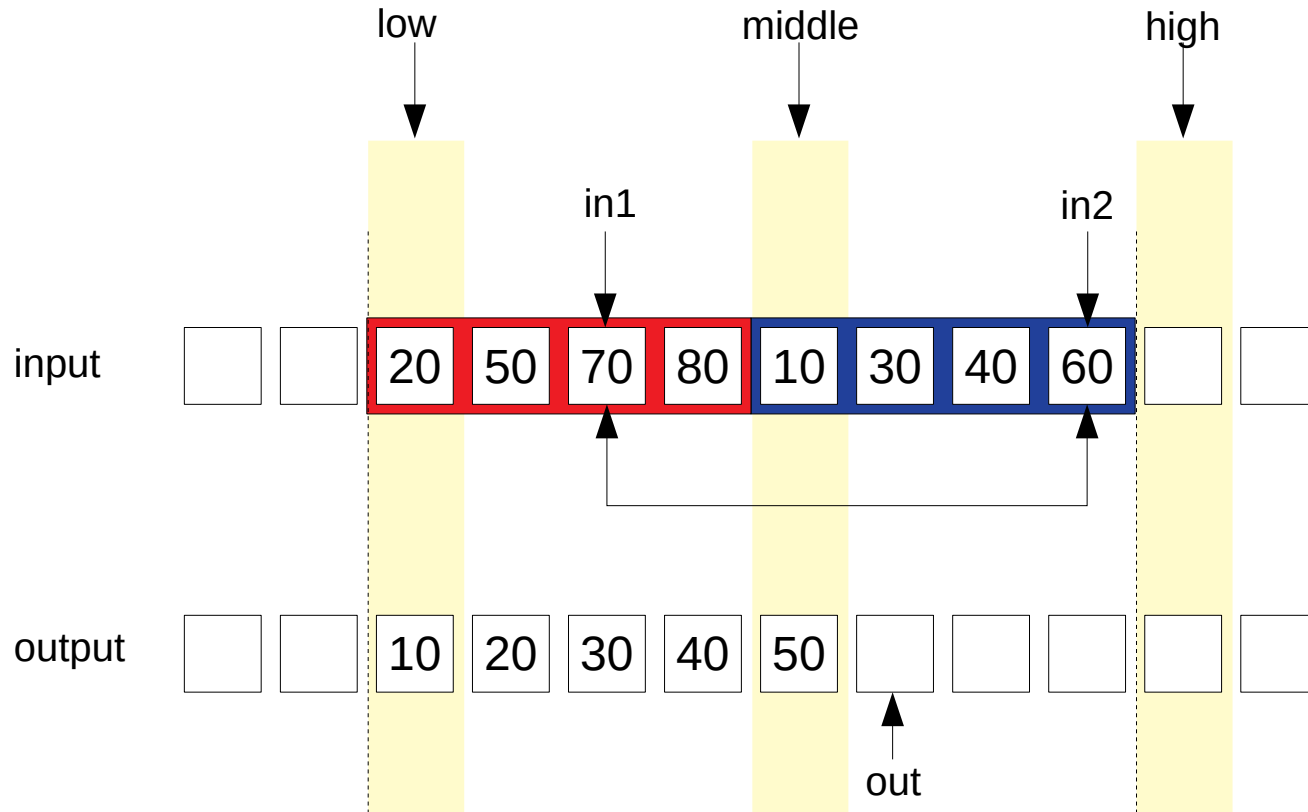
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



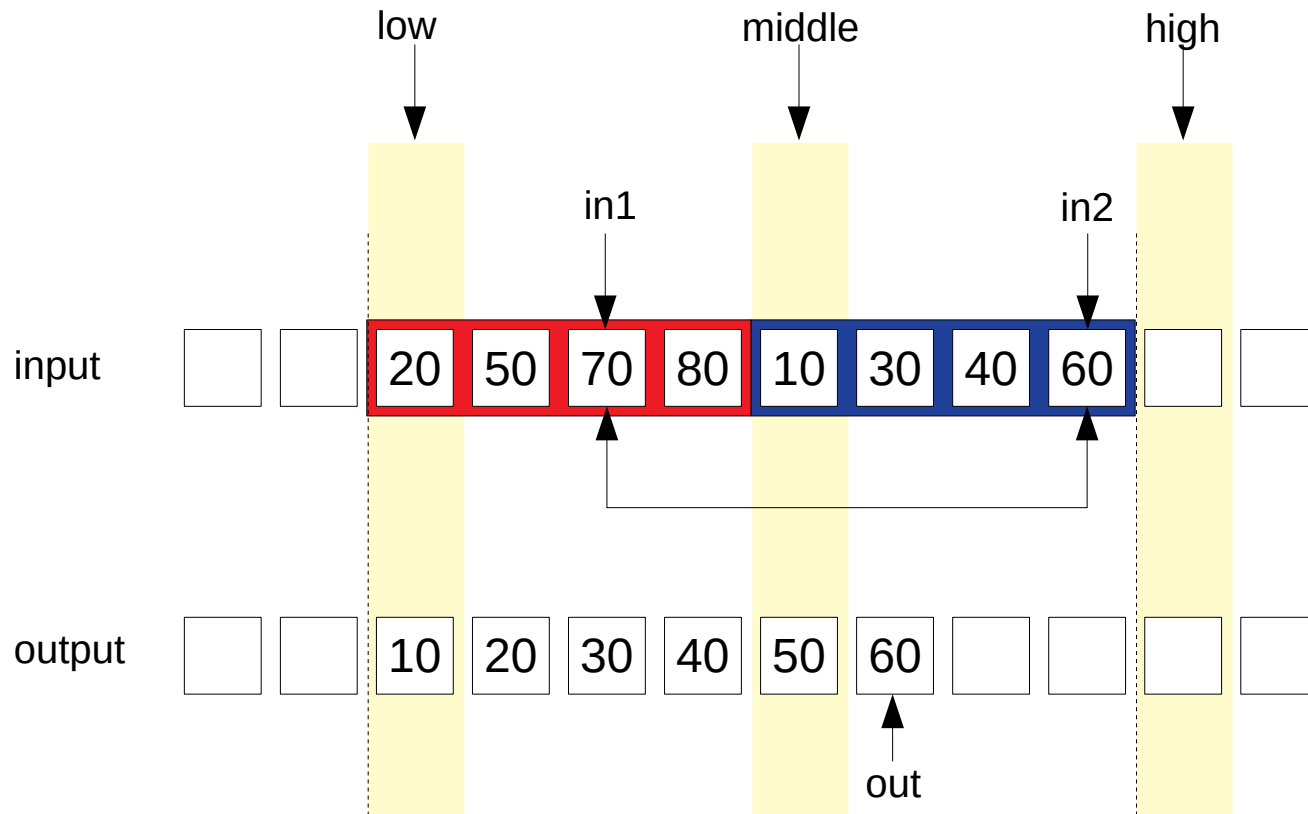
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```


Merge susedných častí poľa



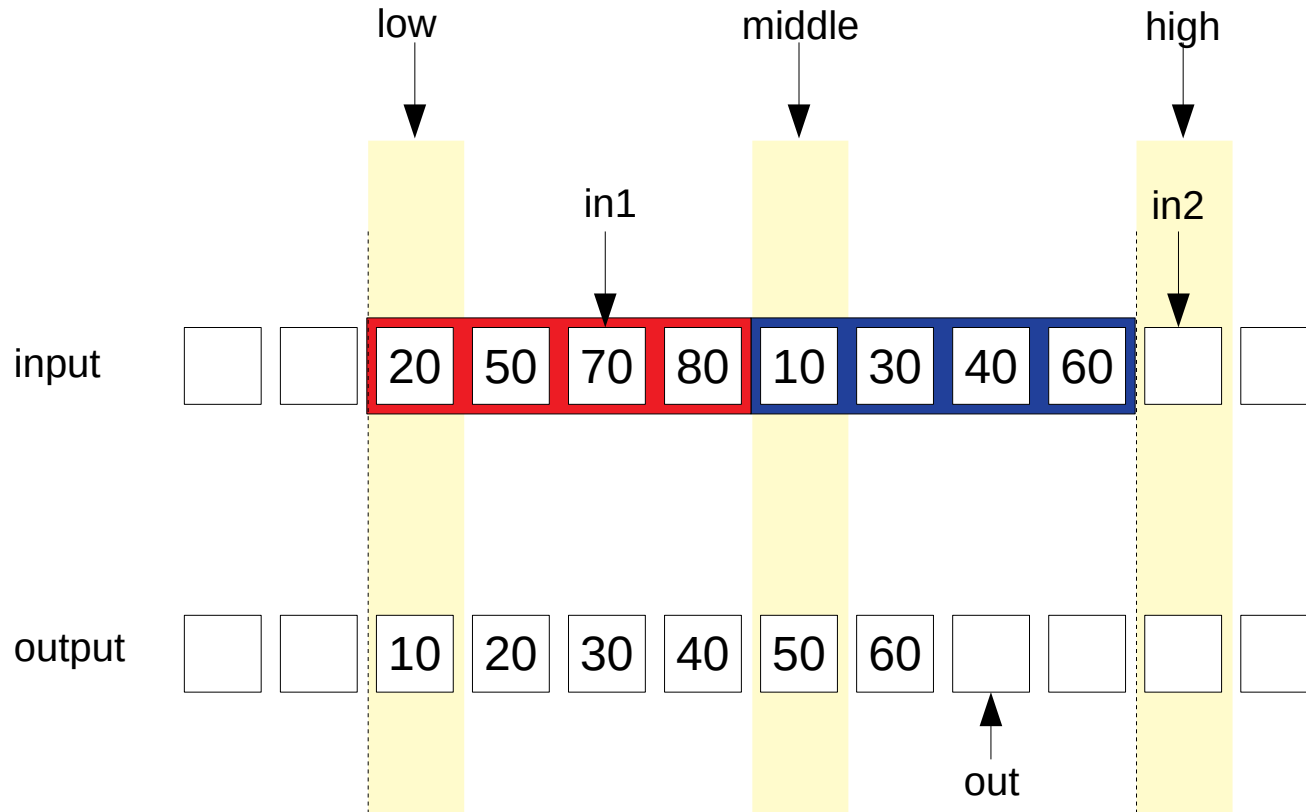
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



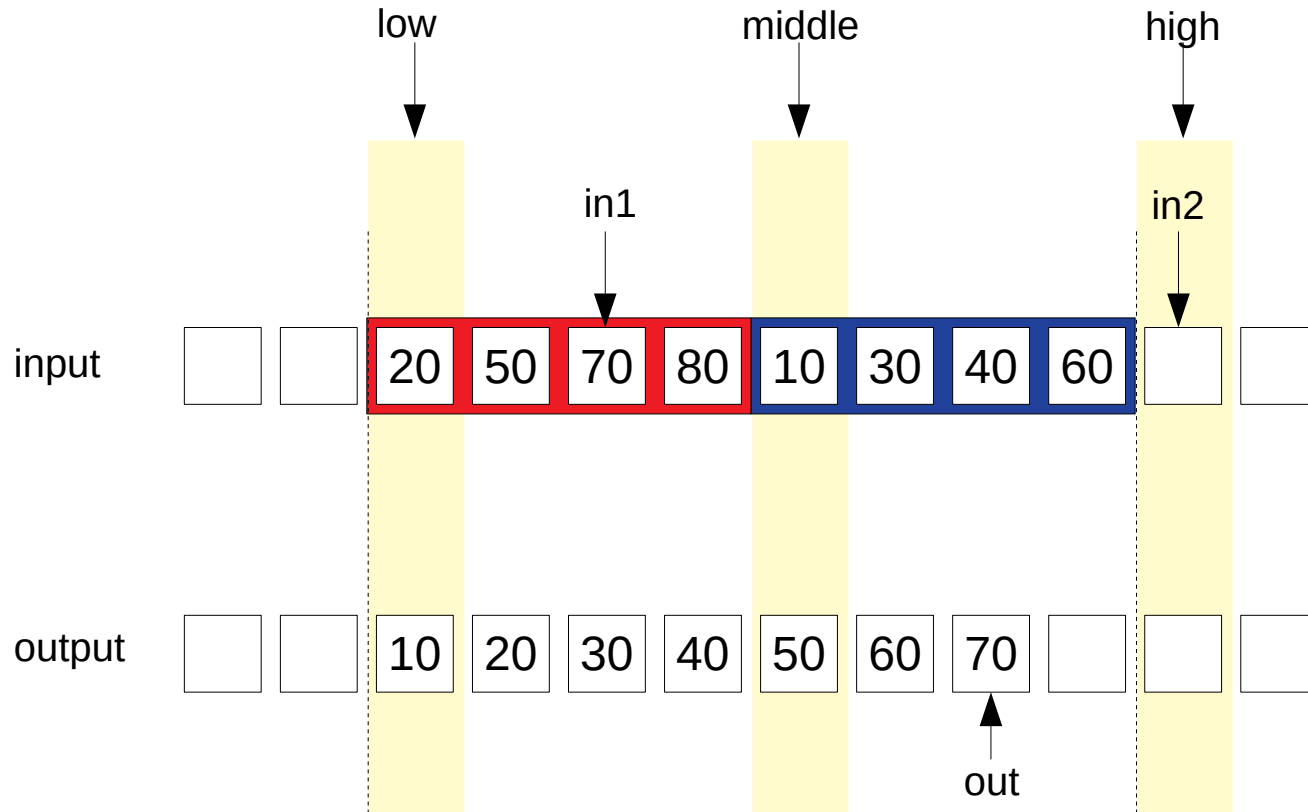
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}
```

Merge susedných častí poľa



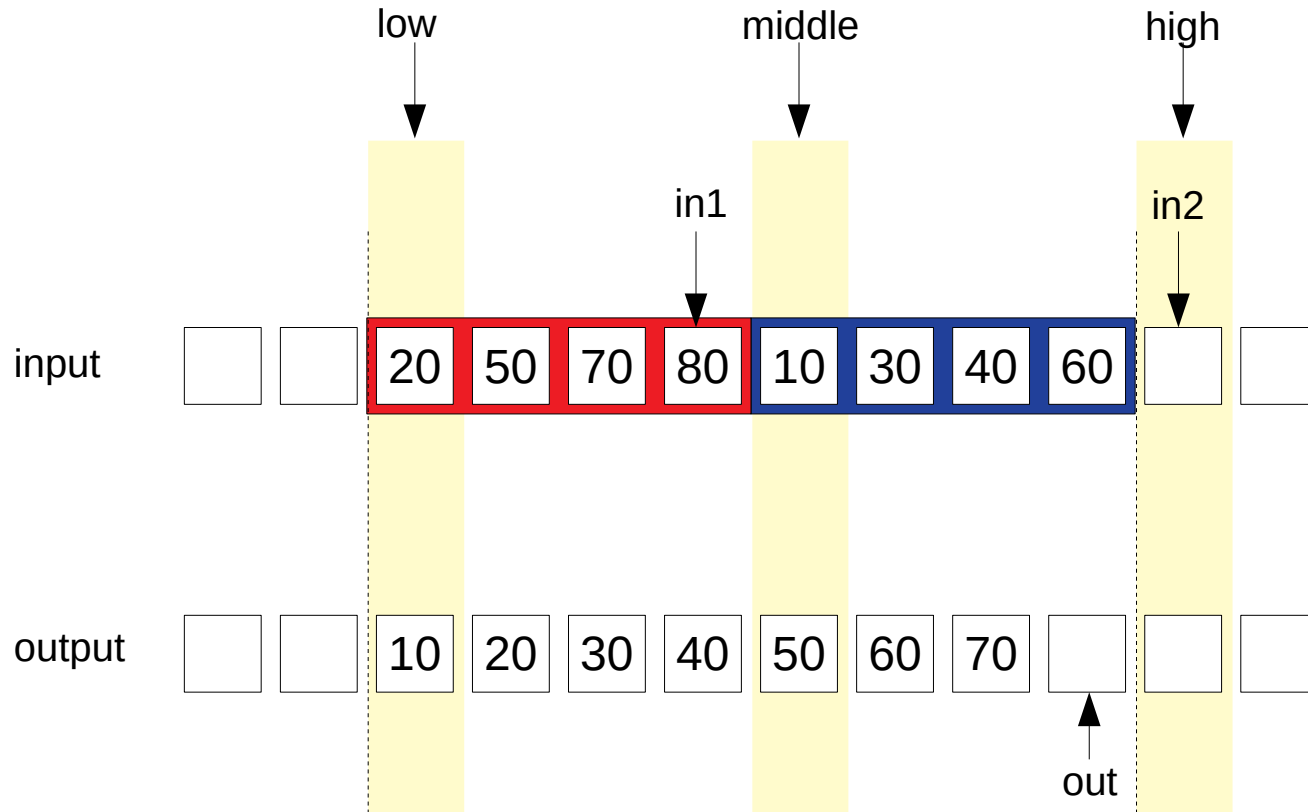
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}  
while(in1 < middle) {  
    output[out] = input[in1]  
    out++  
    in1++  
}
```

Merge susedných častí poľa



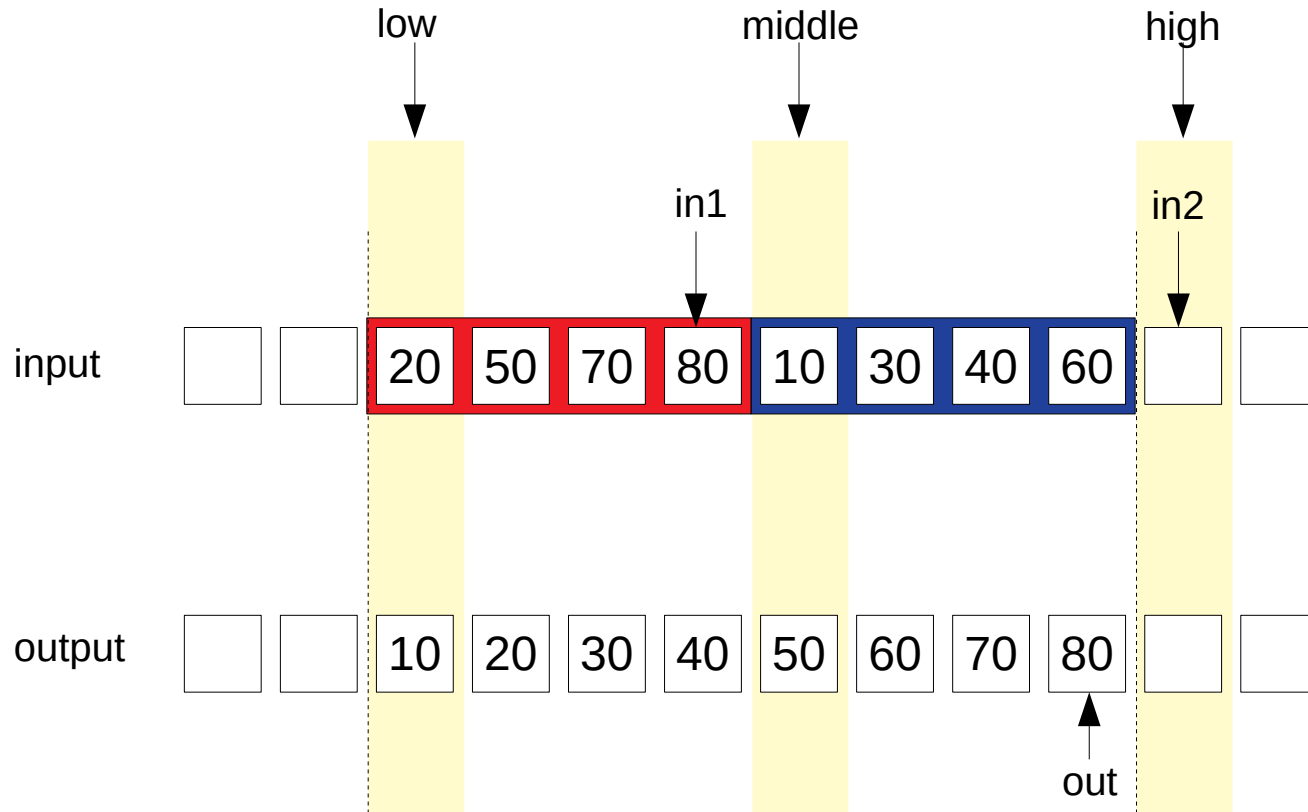
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}  
while(in1 < middle) {  
    output[out] = input[in1]  
    out++  
    in1++  
}
```

Merge susedných častí poľa



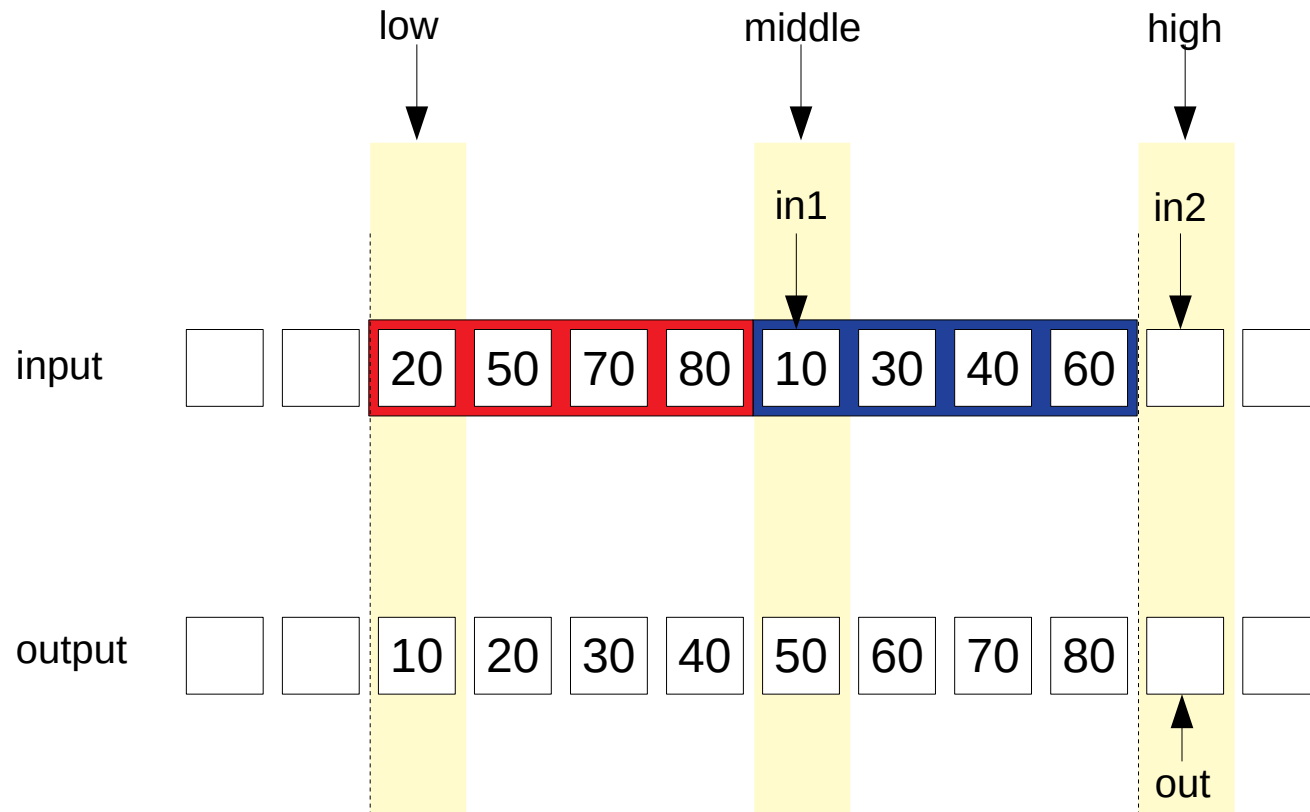
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}  
while(in1 < middle) {  
    output[out] = input[in1]  
    out++  
    in1++  
}
```

Merge susedných častí poľa



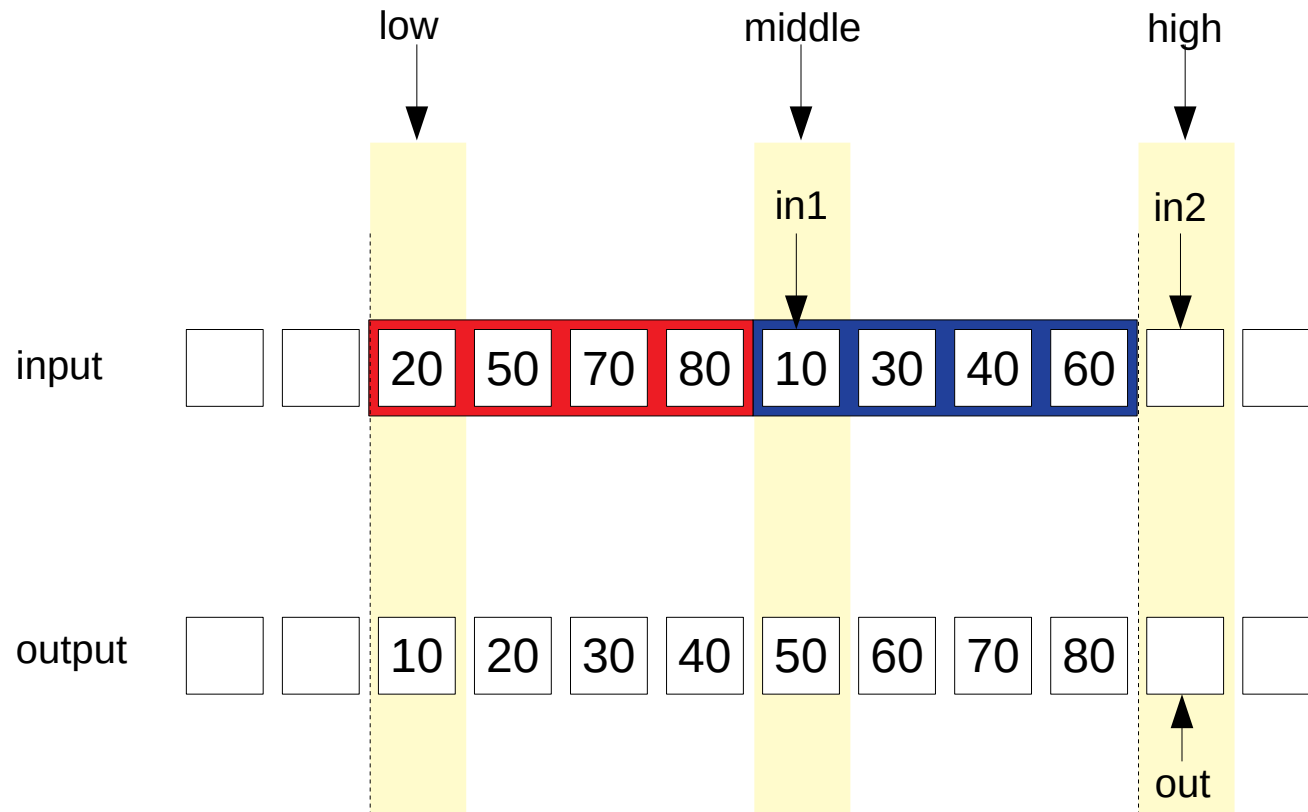
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}  
while(in1 < middle) {  
    output[out] = input[in1]  
    out++  
    in1++  
}
```

Merge susedných častí poľa



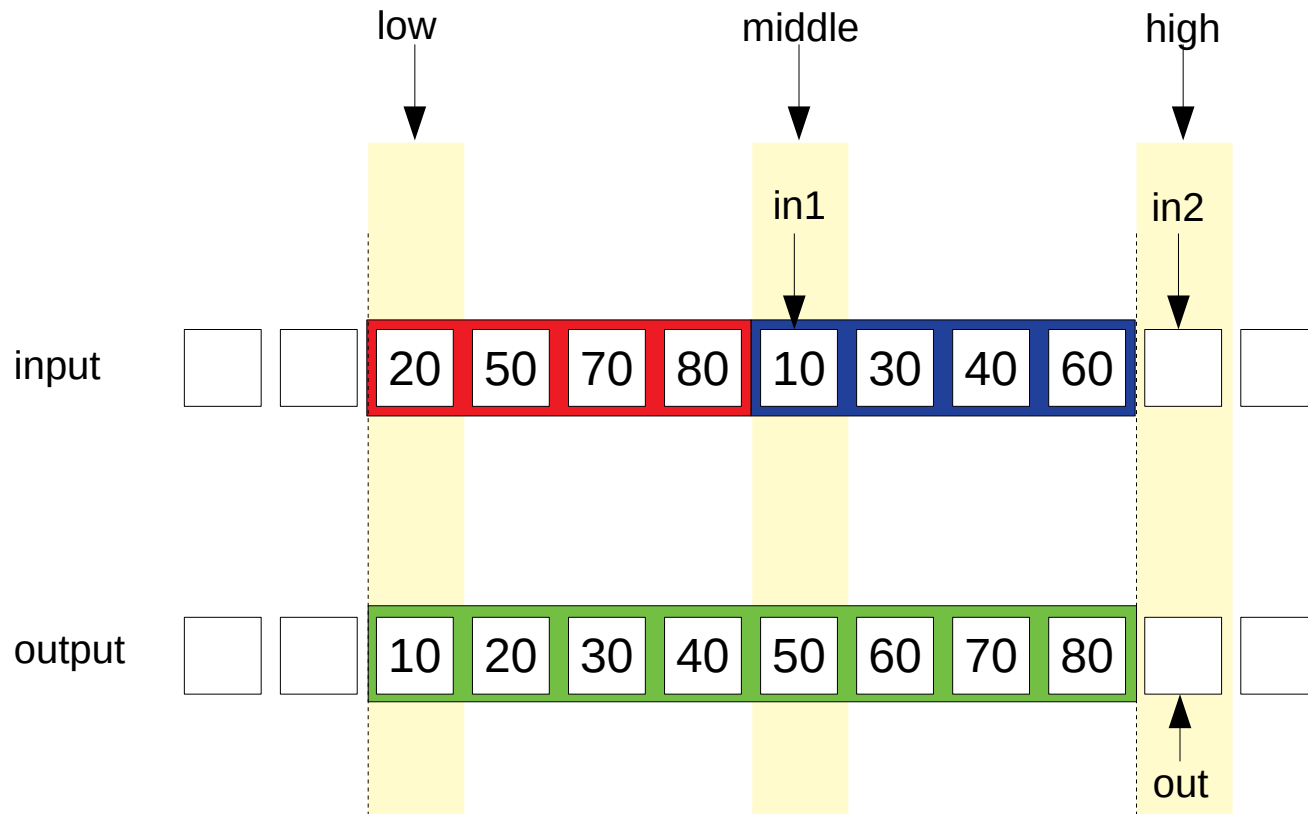
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}  
while(in1 < middle) {  
    output[out] = input[in1]  
    out++  
    in1++  
}
```

Merge susedných častí poľa



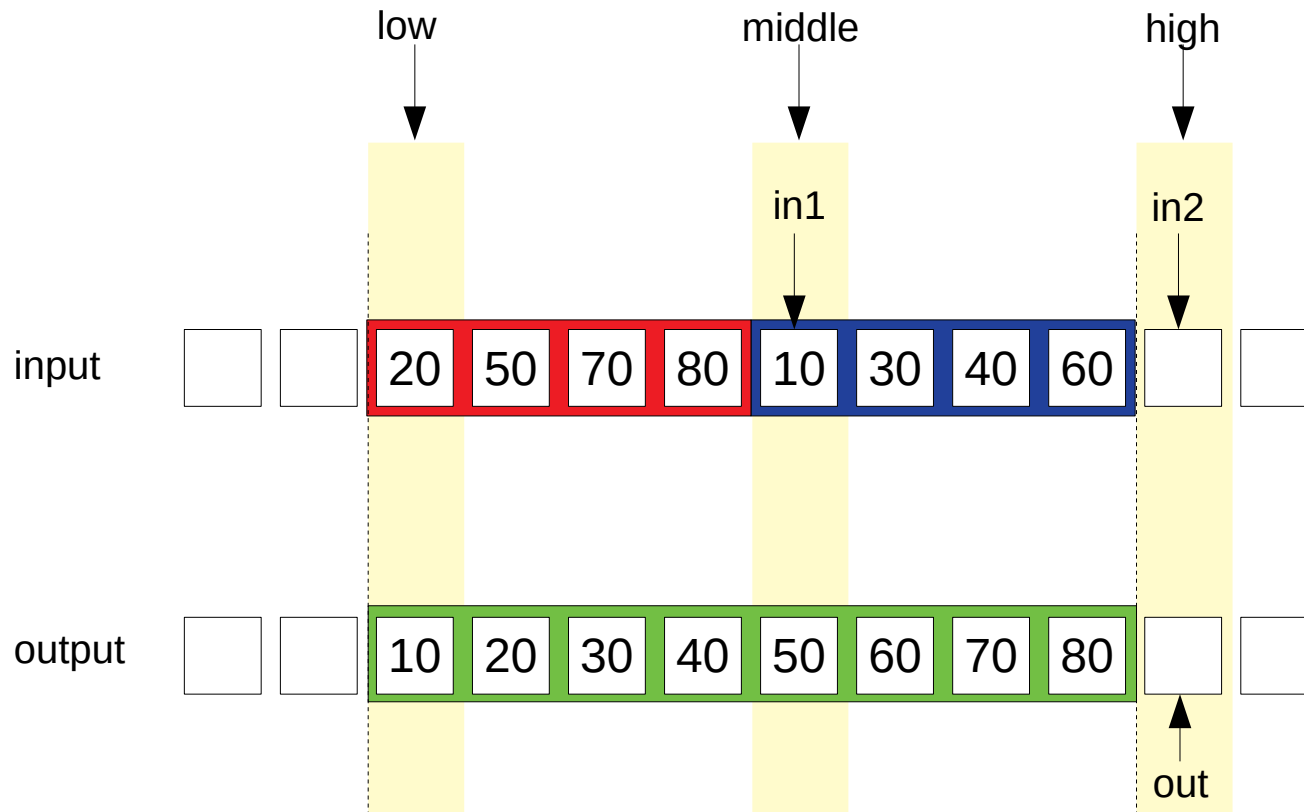
```
while(in1 < middle and in2 < high) {  
    if(input[in1] <= input[in2]) {  
        output[out] = input[in1]  
        out++  
        in1++  
    }  
    else {  
        output[out] = input[in2]  
        out++  
        in2++  
    }  
}  
while(in1 < middle) {  
    output[out] = input[in1]  
    out++  
    in1++  
}  
while(in2 < high) {  
    output[out] = input[in2]  
    out++  
    in2++  
}
```


Merge susedných častí poľa



```
while(in1 < middle and in2 < high) {
    if(input[in1] <= input[in2]) {
        output[out] = input[in1]
        out++
        in1++
    }
    else {
        output[out] = input[in2]
        out++
        in2++
    }
}
while(in1 < middle) {
    output[out] = input[in1]
    out++
    in1++
}
while(in2 < high) {
    output[out] = input[in2]
    out++
    in2++
}
```

Merge susedných častí poľa



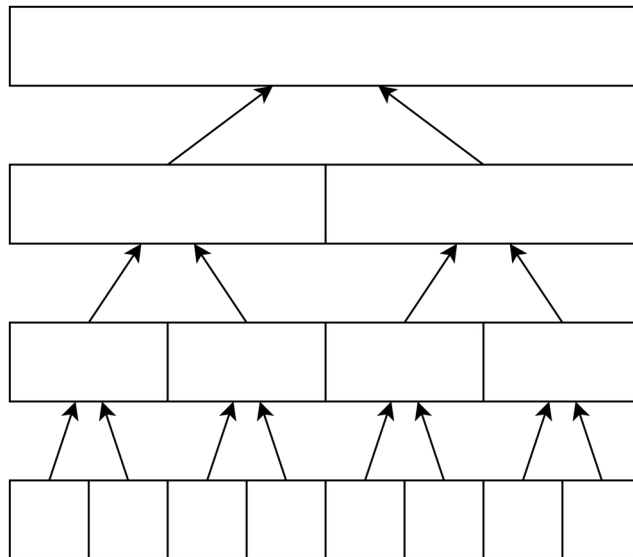
stabilné
triedenie

```

while(in1 < middle and in2 < high) {
    if(input[in1] <= input[in2]) {
        output[out] = input[in1]
        out++
        in1++
    }
    else {
        output[out] = input[in2]
        out++
        in2++
    }
}
while(in1 < middle) {
    output[out] = input[in1]
    out++
    in1++
}
while(in2 < high) {
    output[out] = input[in2]
    out++
    in2++
}

```

Merge sort – bottom up



Merge sort – bottom up

hlavné
pole

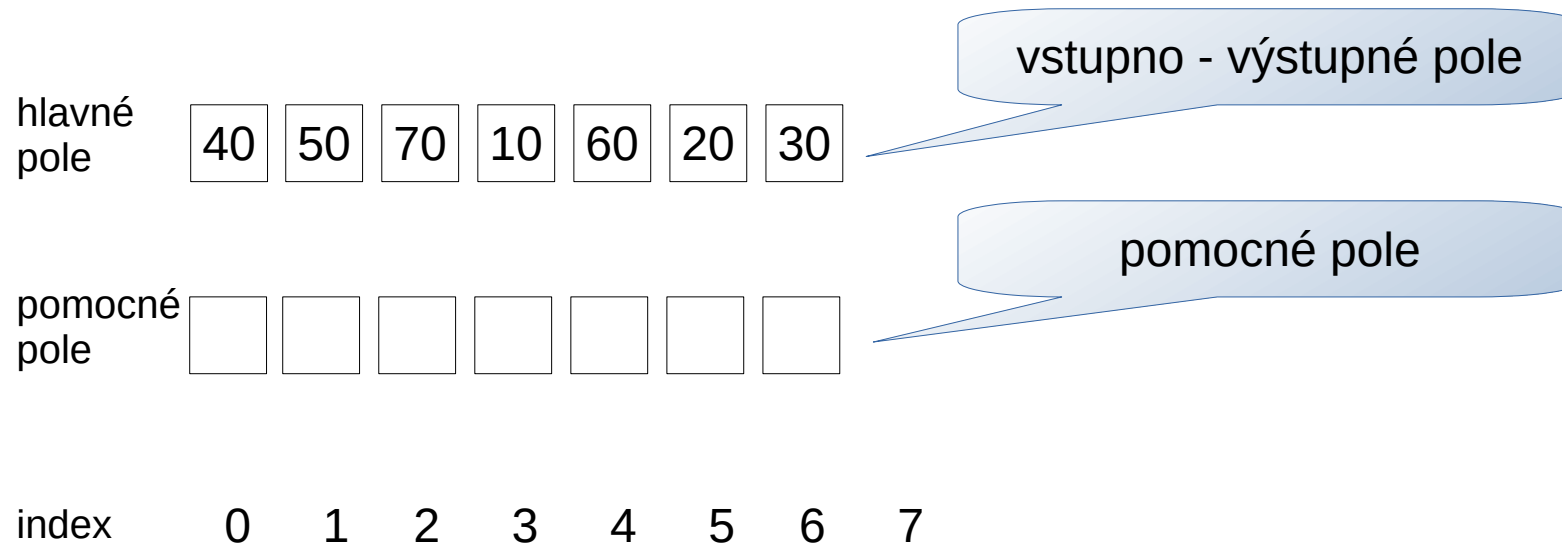
40	50	70	10	60	20	30
----	----	----	----	----	----	----

vstupno - výstupné pole

index

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Merge sort – bottom up



Merge sort – bottom up

hlavné
pole

40	50	70	10	60	20	30

pomocné
pole

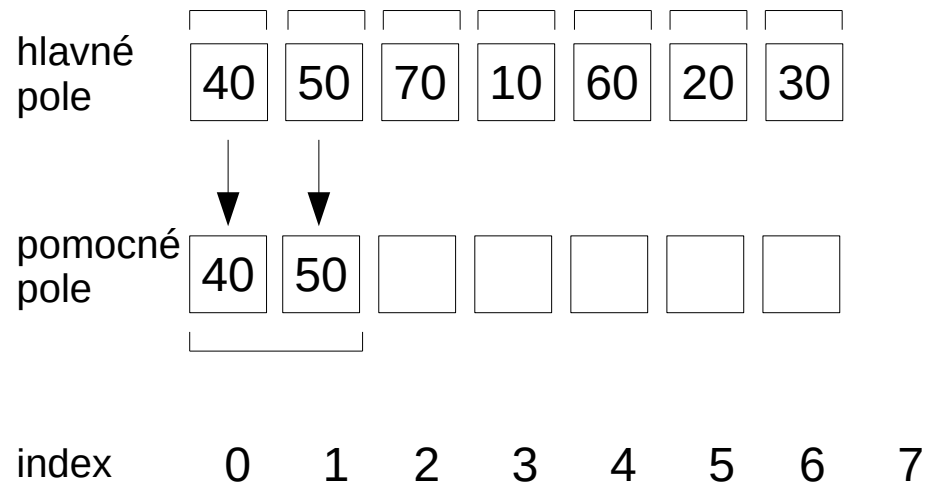
--	--	--	--	--	--	--

index

0	1	2	3	4	5	6	7
---	---	---	---	---	---	---	---

Merge sort – bottom up

step = 1
merge(0,1,2)

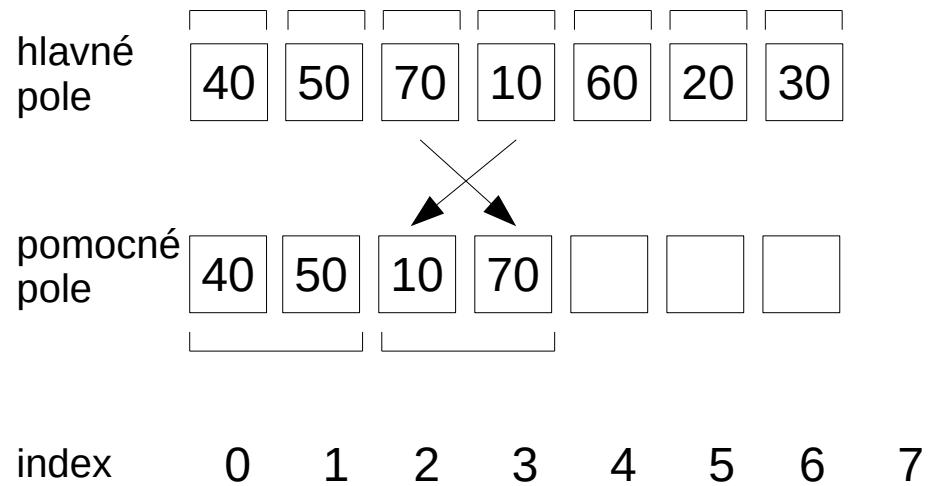


Merge sort – bottom up

step = 1

merge(0,1,2)

merge(2,3,4)



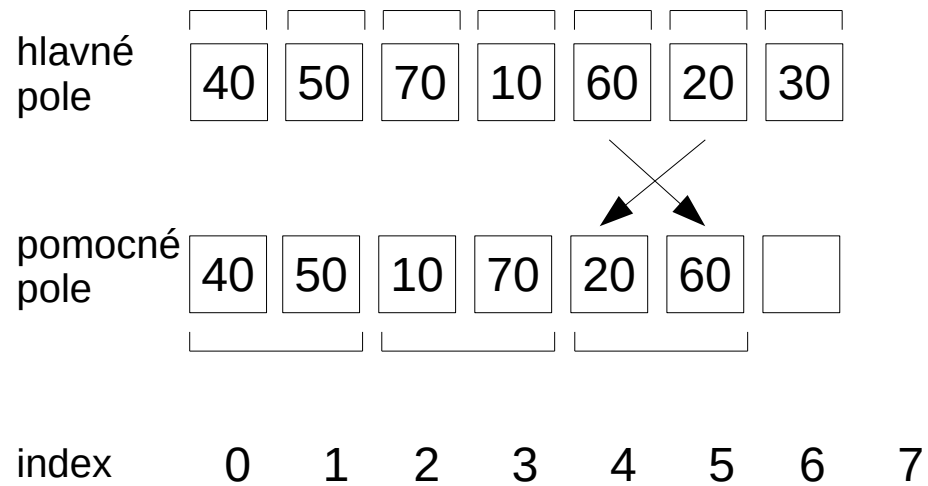
Merge sort – bottom up

step = 1

merge(0,1,2)

merge(2,3,4)

merge(4,5,6)



Merge sort – bottom up

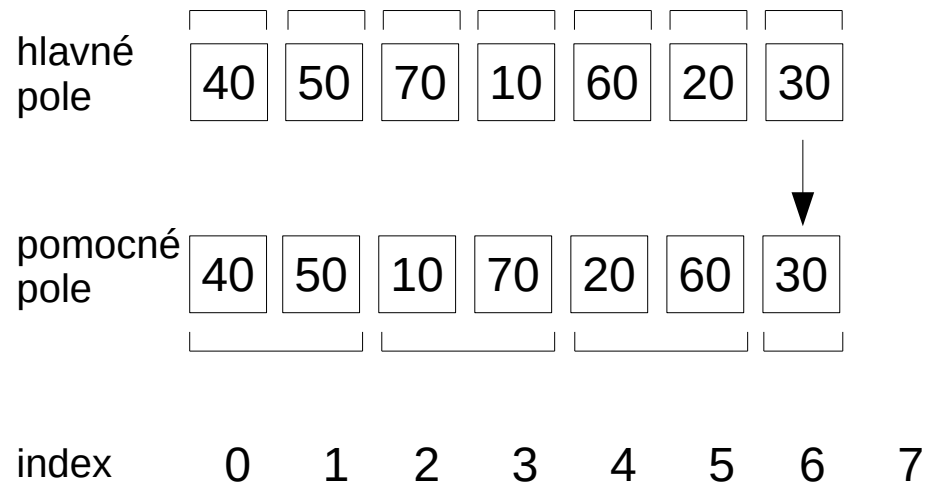
step = 1

merge(0,1,2)

merge(2,3,4)

merge(4,5,6)

merge(6,7,7)



Merge sort – bottom up

step = 1

merge(0,1,2)

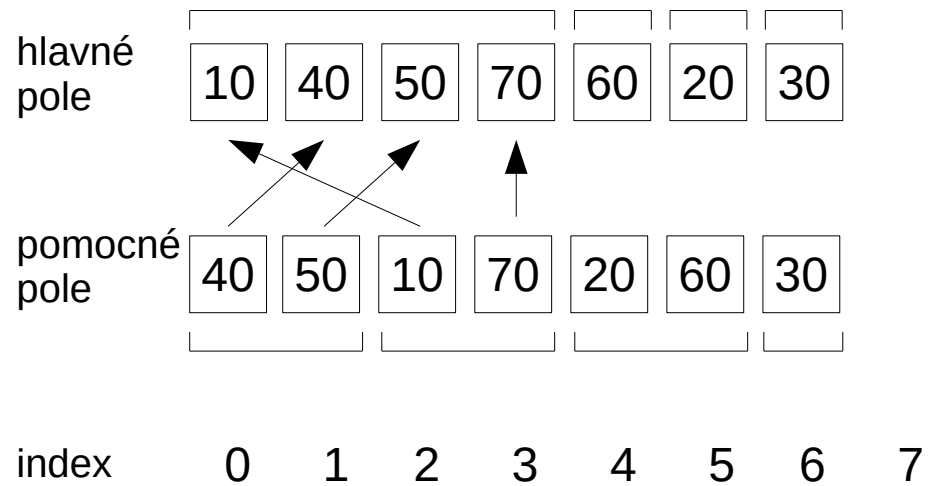
merge(2,3,4)

merge(4,5,6)

merge(6,7,7)

step = 2

merge(0, 2, 4)



Merge sort – bottom up

step = 1

merge(0,1,2)

merge(2,3,4)

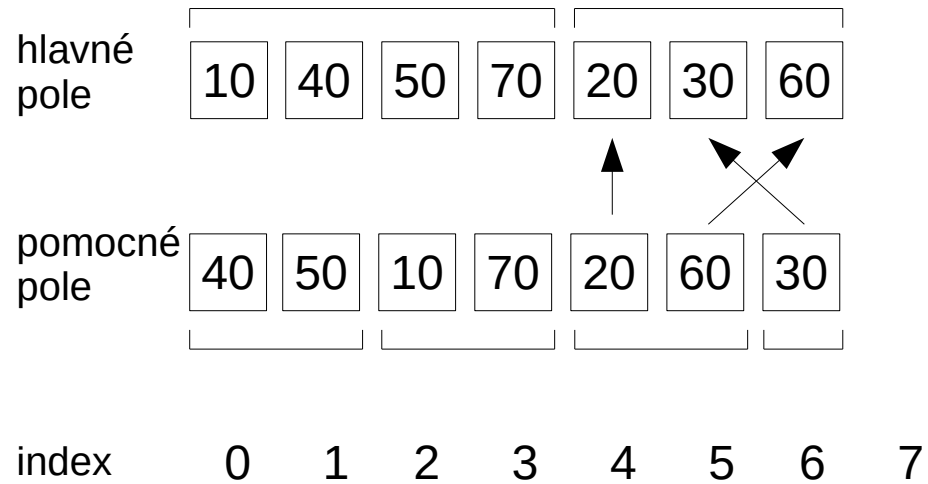
merge(4,5,6)

merge(6,7,7)

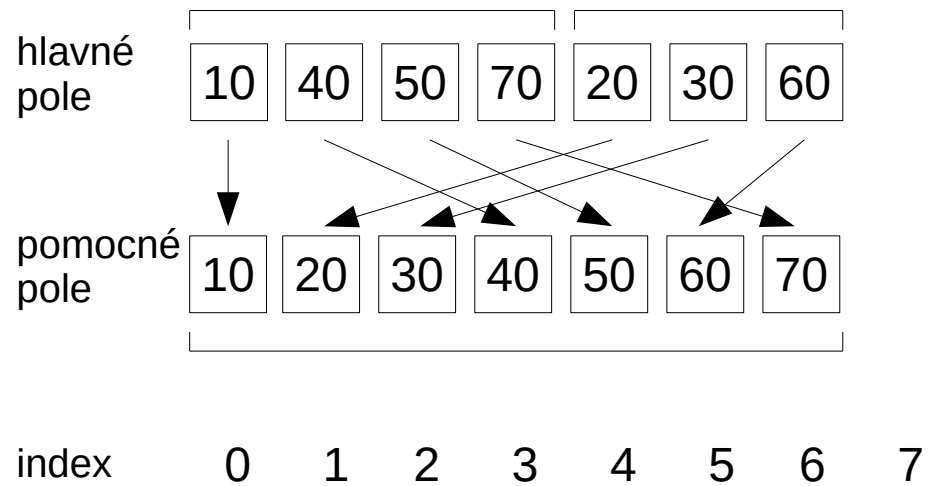
step = 2

merge(0, 2, 4)

merge(4,6,7)



Merge sort – bottom up



step = 1

merge(0,1,2)

merge(2,3,4)

merge(4,5,6)

merge(6,7,7)

step = 2

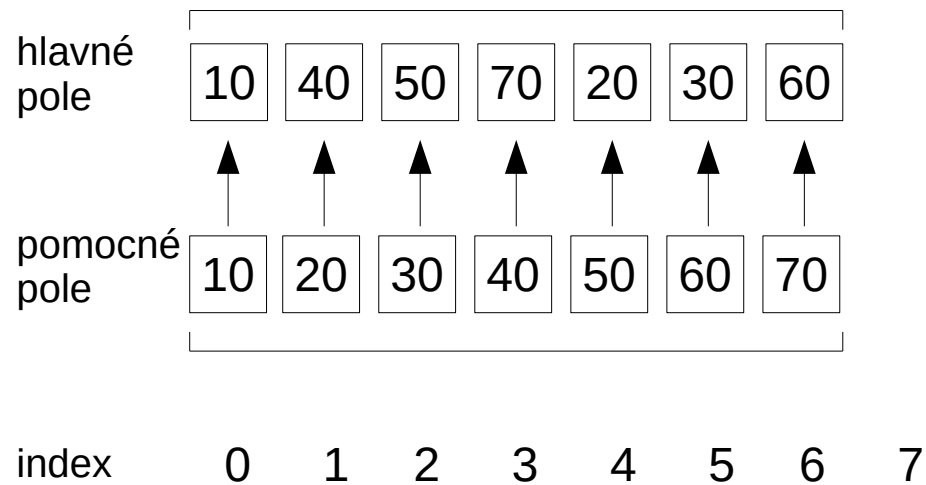
merge(0, 2, 4)

merge(4,6,7)

step = 4

merge(0,4, 7)

Merge sort – bottom up



step = 1

merge(0,1,2)

merge(2,3,4)

merge(4,5,6)

merge(6,7,7)

step = 2

merge(0, 2, 4)

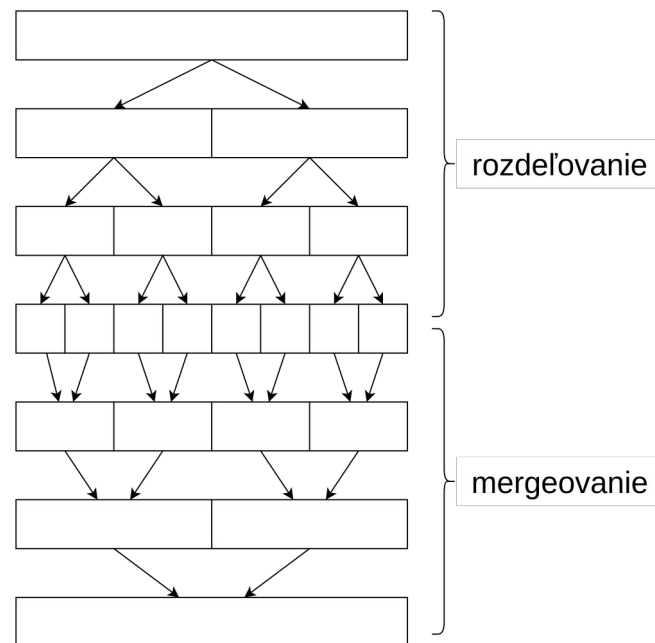
merge(4,6,7)

step = 4

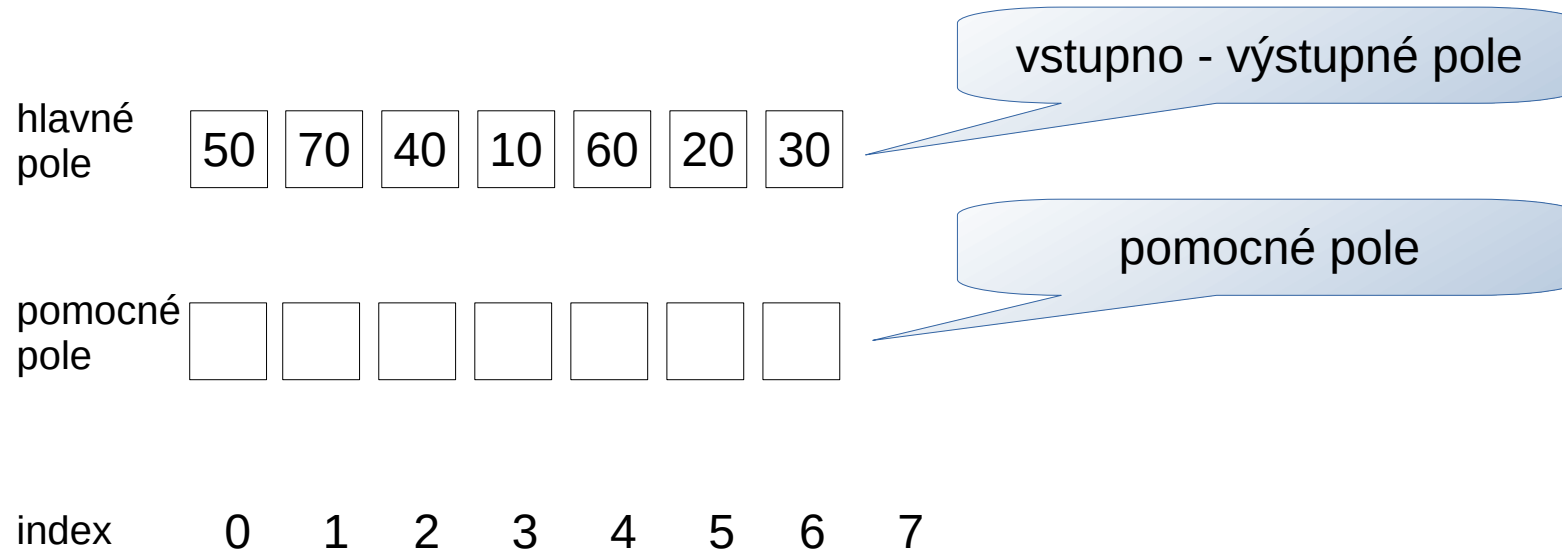
merge(0,4, 7)

copy (ak je potrebné – nepárny počet
kopírovaní z jedného pola do druhého)

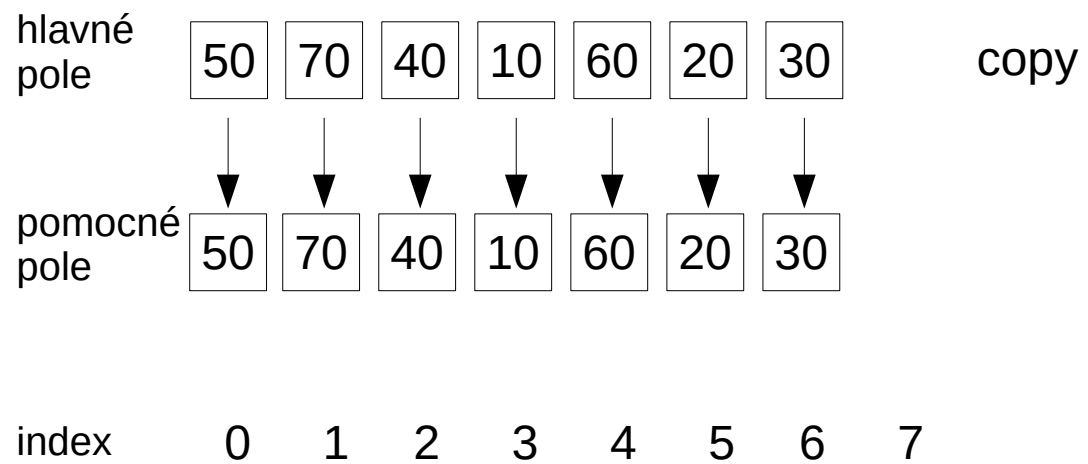
Merge sort – top down



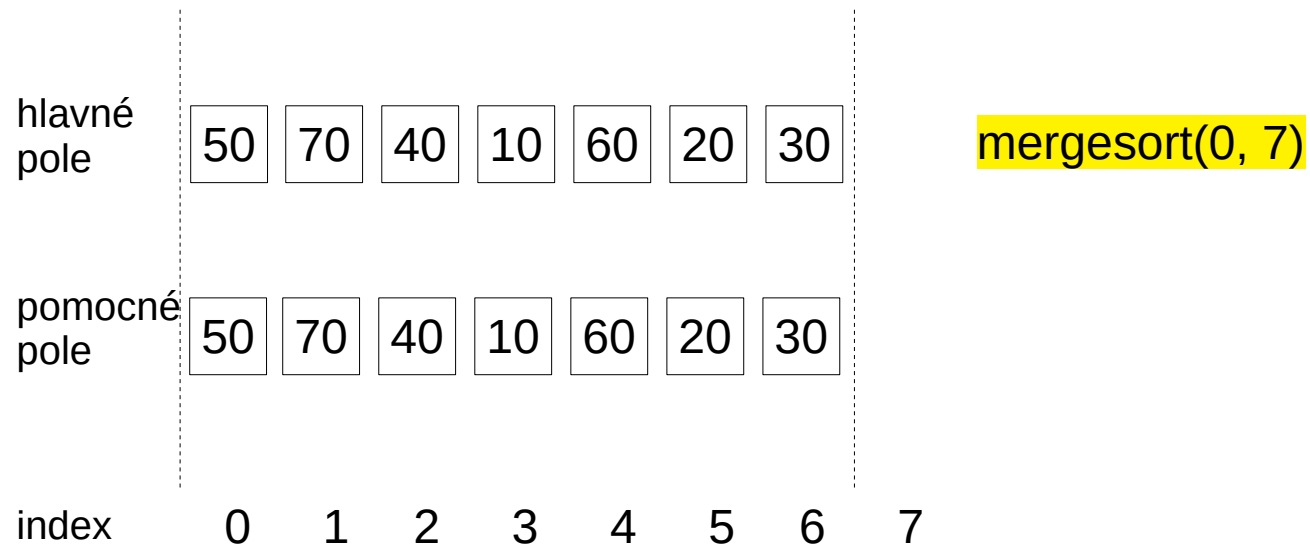
Merge sort – top down



Merge sort – top down

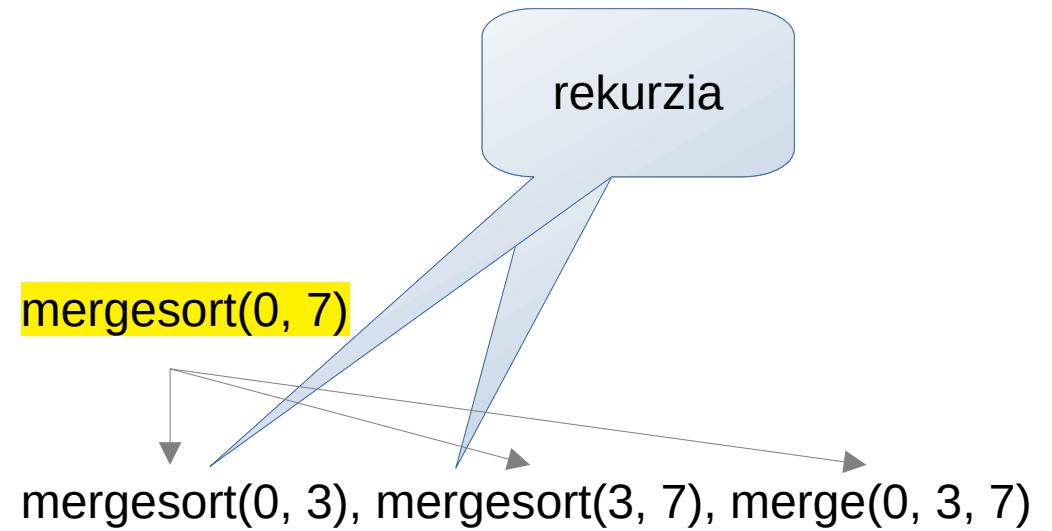


Merge sort – top down

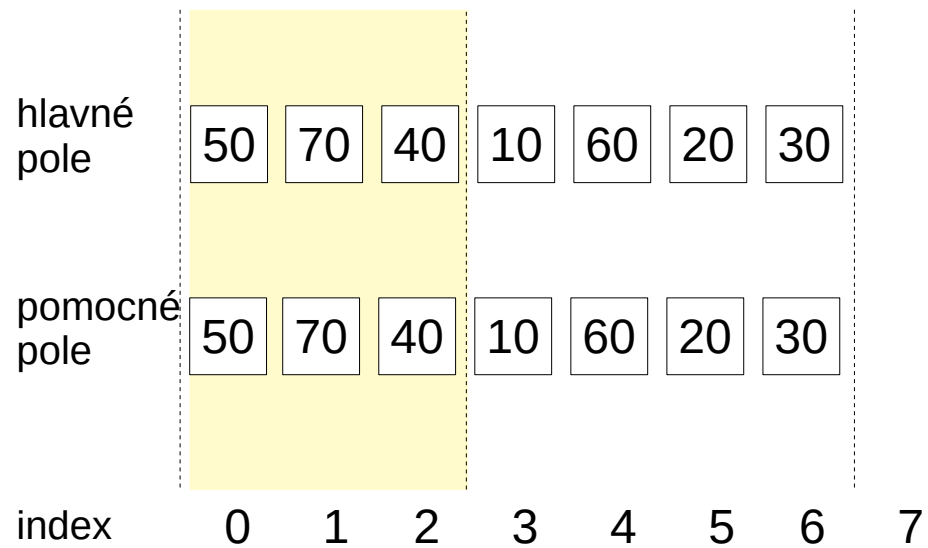


Merge sort – top down

hlavné pole	50	70	40	10	60	20	30	
pomocné pole	50	70	40	10	60	20	30	
index	0	1	2	3	4	5	6	7



Merge sort – top down



`mergesort(0, 7)`

Diagram showing the recursive calls for the merge sort algorithm. The initial call `mergesort(0, 7)` branches into three recursive calls: `mergesort(0, 3)`, `mergesort(3, 7)`, and `merge(0, 3, 7)`.

Merge sort – top down

hlavné pole	50	70	40	10	60	20	30	
pomocné pole	50	70	40	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), mergesort(1, 3), merge(0, 1, 3)

Merge sort – top down

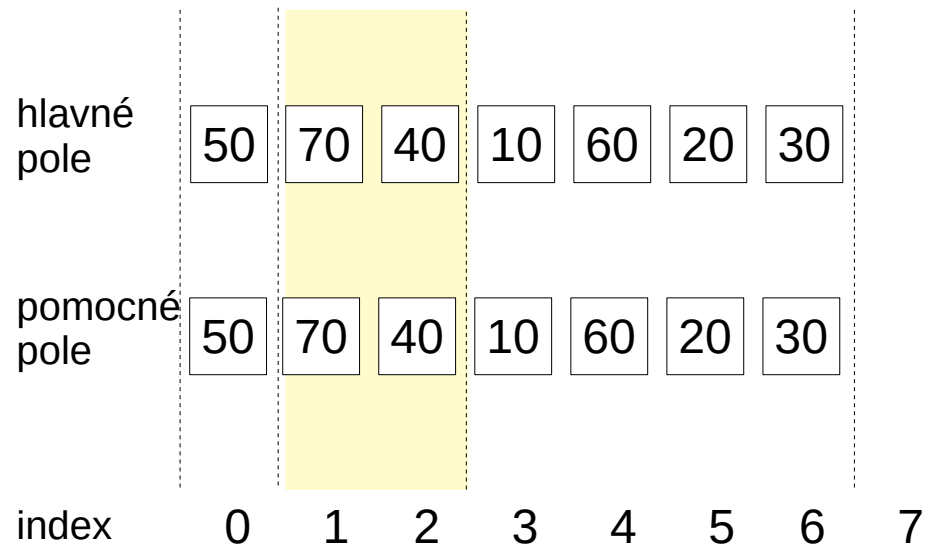
hlavné pole	50	70	40	10	60	20	30	
pomocné pole	50	70	40	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), mergesort(1, 3), merge(0, 1, 3)

Merge sort – top down

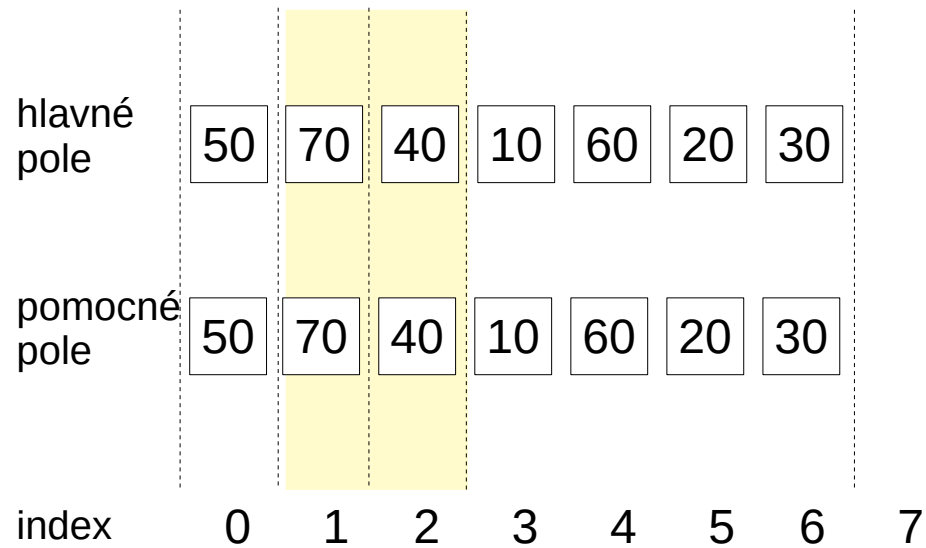


mergesort(0, 7)

↓
mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

↓
mergesort(0, 1), **mergesort(1, 3)**, merge(0, 1, 3)

Merge sort – top down



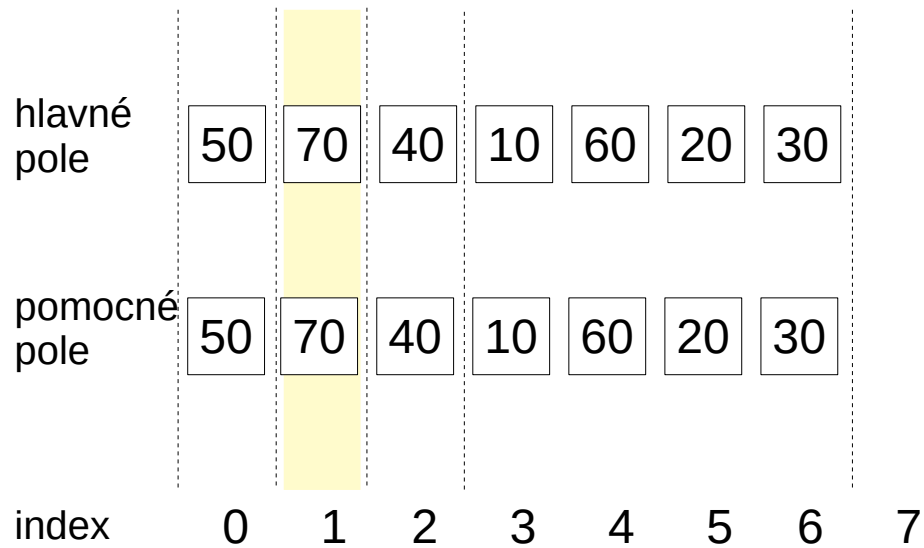
mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), **mergesort(1, 3)**, merge(0, 1, 3)

mergesort(1, 2), mergesort(2, 3), merge(1, 2, 3)

Merge sort – top down



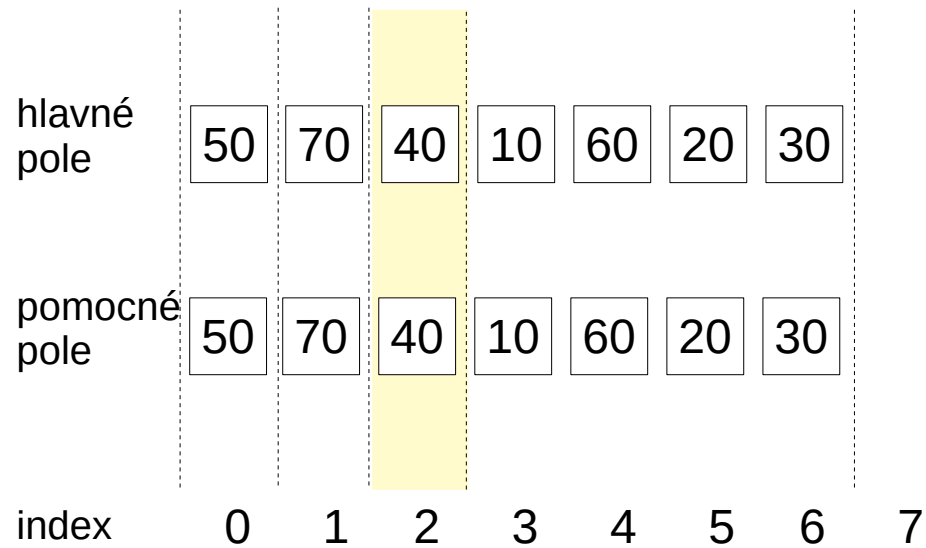
mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), **mergesort(1, 3)**, merge(0, 1, 3)

mergesort(1, 2), mergesort(2, 3), merge(1, 2, 3)

Merge sort – top down



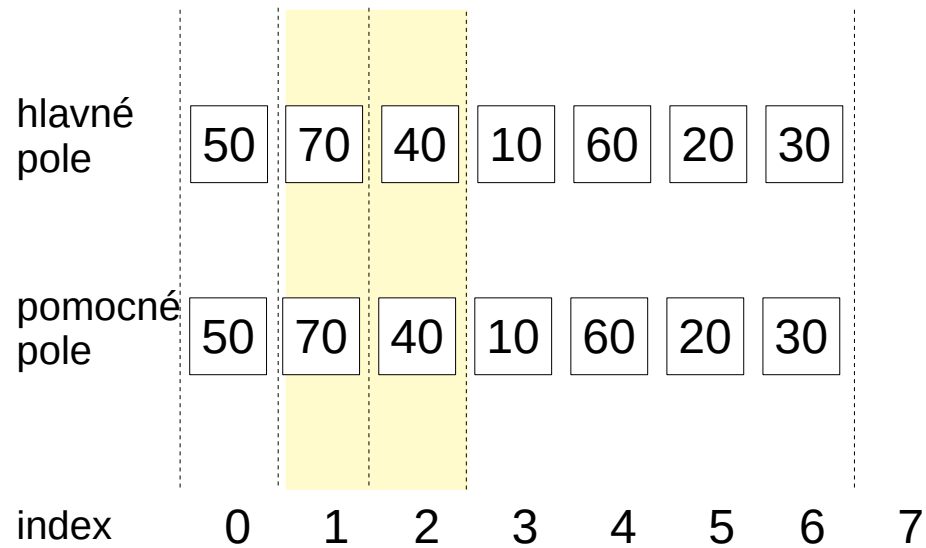
mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), **mergesort(1, 3)**, merge(0, 1, 3)

mergesort(1, 2), **mergesort(2, 3)**, merge(1, 2, 3)

Merge sort – top down



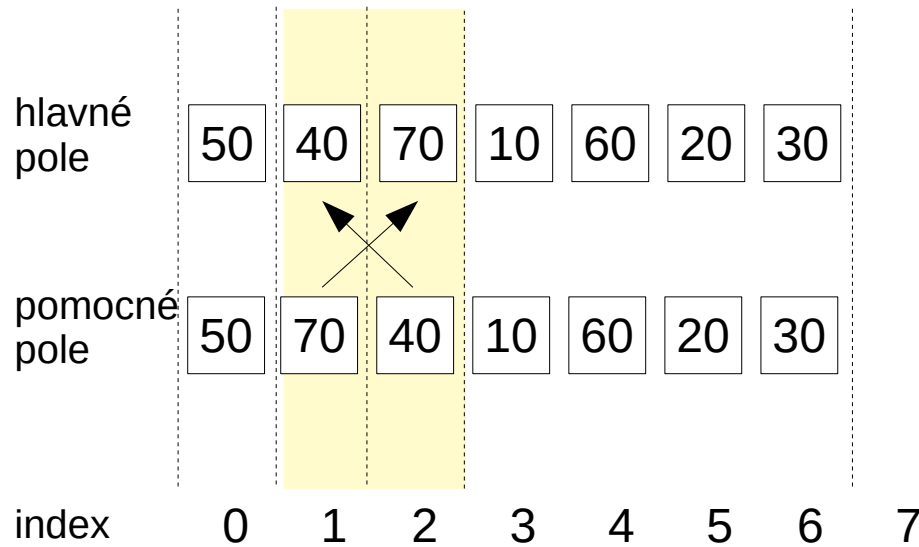
mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), **mergesort(1, 3)**, merge(0, 1, 3)

mergesort(1, 2), mergesort(2, 3), **merge(1, 2, 3)**

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), **mergesort(1, 3)**, merge(0, 1, 3)

mergesort(1, 2), mergesort(2, 3), **merge(1, 2, 3)** ↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	50	70	40	10	60	20	30	
index	0	1	2	3	4	5	6	7

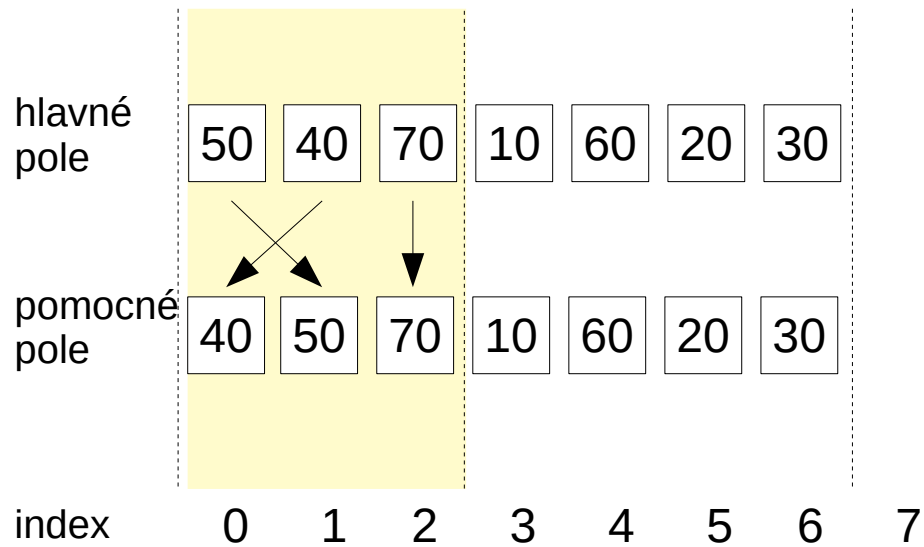
mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), mergesort(1, 3), **merge(0, 1, 3)**

↑
copy

Merge sort – top down



mergesort(0, 7)

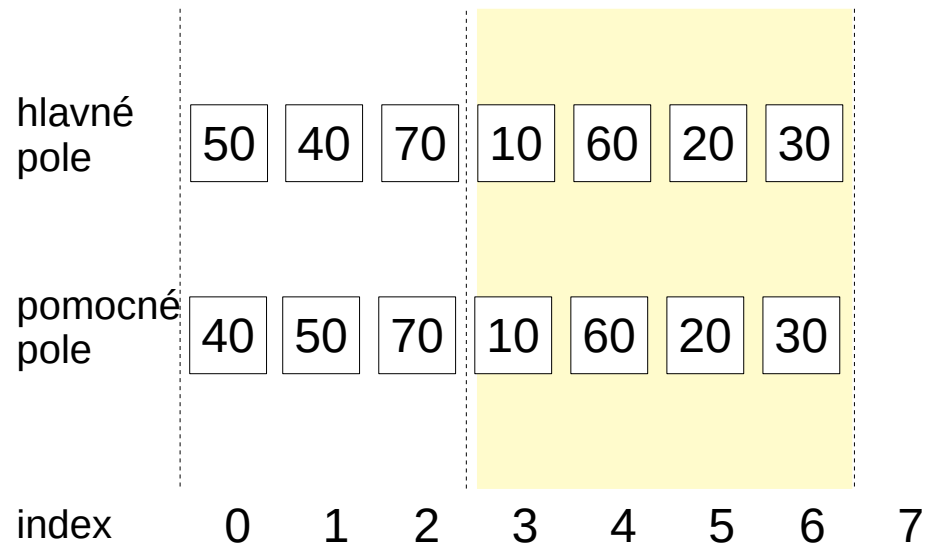
mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(0, 1), mergesort(1, 3), **merge(0, 1, 3)**

↓ copy

↑ copy

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

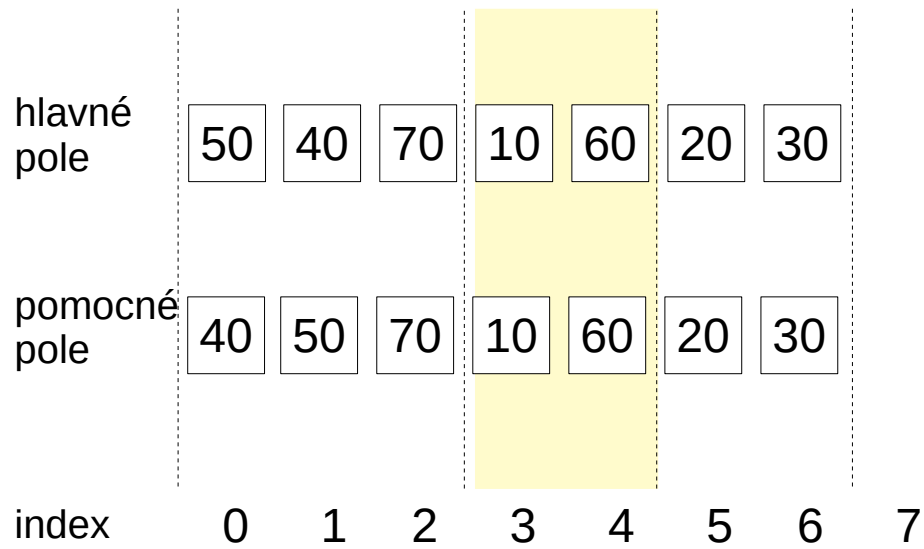
mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

↓ copy

↑ copy

Merge sort – top down



mergesort(0, 7)

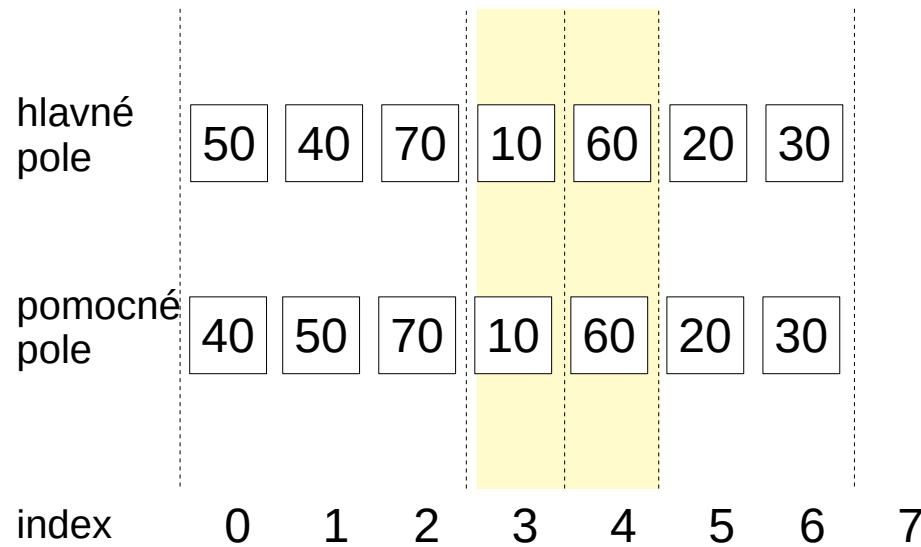
mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

↓ copy

↑ copy

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

mergesort(3, 4), mergesort(4, 5), merge(3, 4, 5)

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

mergesort(3, 4), mergesort(4, 5), merge(3, 4, 5)

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	60	20	30	
index	0	1	2	3	4	5	6	7

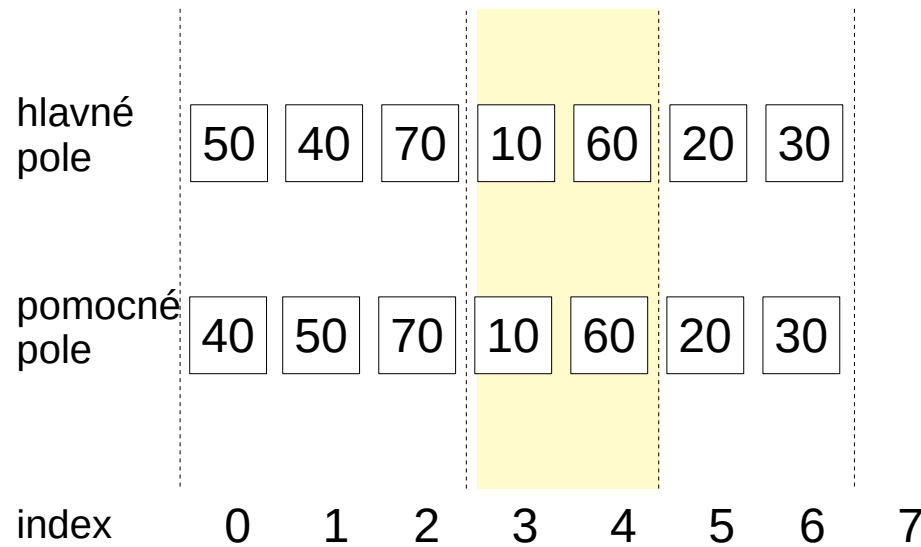
mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7) ↓ copy

mergesort(3, 4), **mergesort(4, 5)**, merge(3, 4, 5) ↑ copy

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

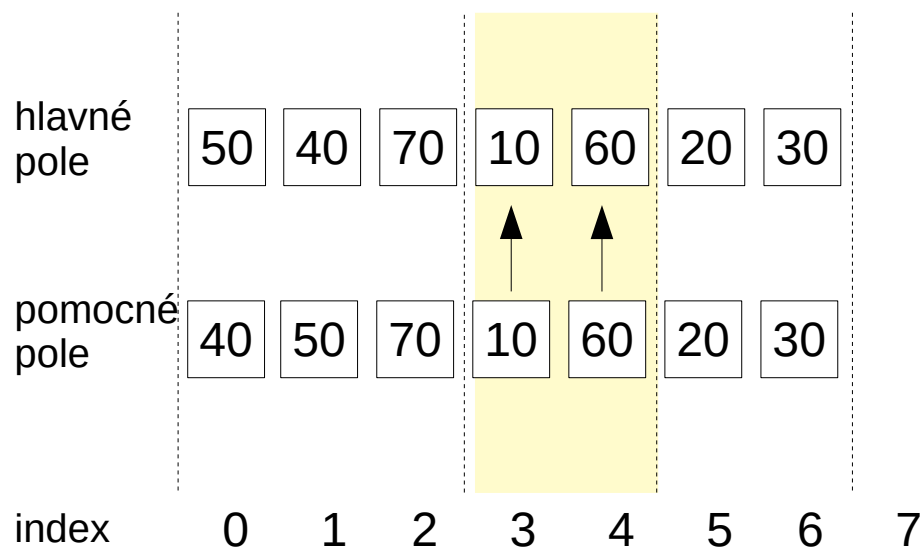
mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

mergesort(3, 4), mergesort(4, 5), **merge(3, 4, 5)**

↓ copy

↑ copy

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

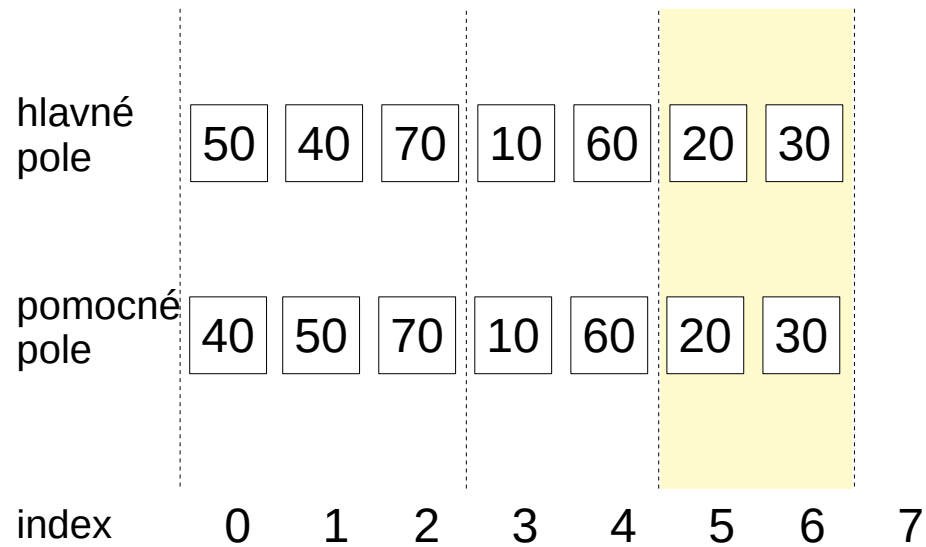
mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

mergesort(3, 4), mergesort(4, 5), **merge(3, 4, 5)**

↓ copy

↑ copy

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), **mergesort(5, 7)**, merge(3, 5, 7)

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

mergesort(5, 6), mergesort(6, 7), merge(5, 6, 7)

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), **mergesort(5, 7)**, merge(3, 5, 7)

mergesort(5, 6), mergesort(6, 7), merge(5, 6, 7)

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30
pomocné pole	40	50	70	10	60	20	30
index	0	1	2	3	4	5	6

mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), **mergesort(5, 7)**, merge(3, 5, 7) ↓ copy

mergesort(5, 6), **mergesort(6, 7)**, merge(5, 6, 7) ↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	60	20	30	
index	0	1	2	3	4	5	6	7

mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), merge(0, 3, 7)

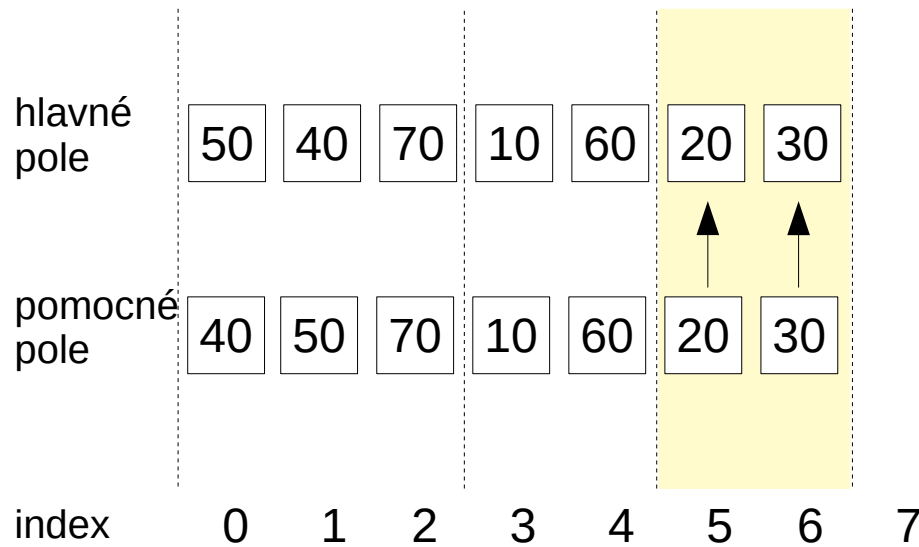
mergesort(3, 5), mergesort(5, 7), merge(3, 5, 7)

mergesort(5, 6), mergesort(6, 7), merge(5, 6, 7)

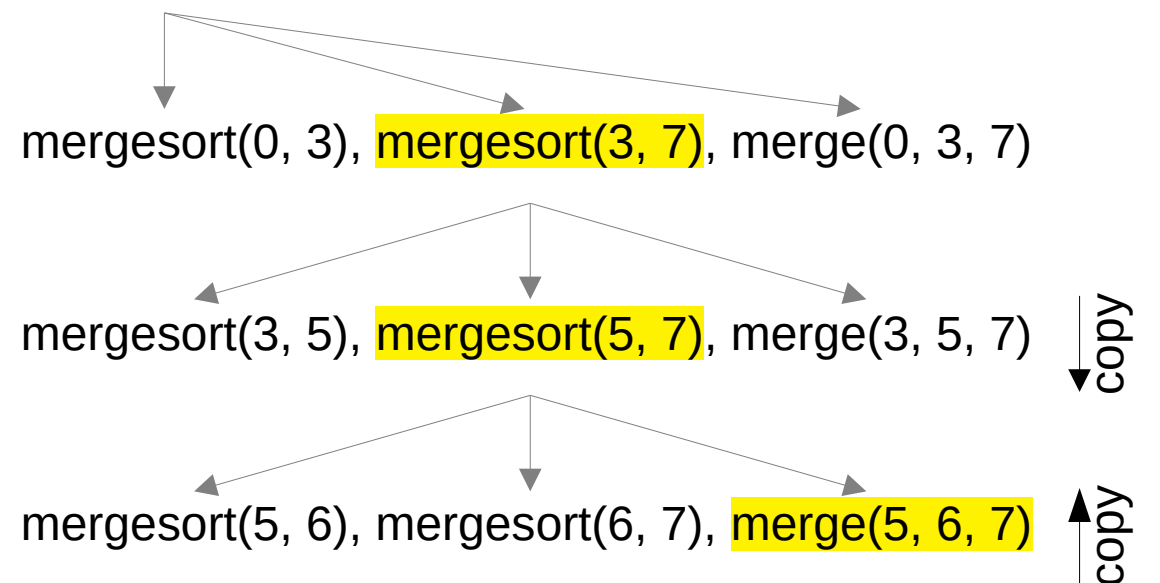
↓ copy

↑ copy

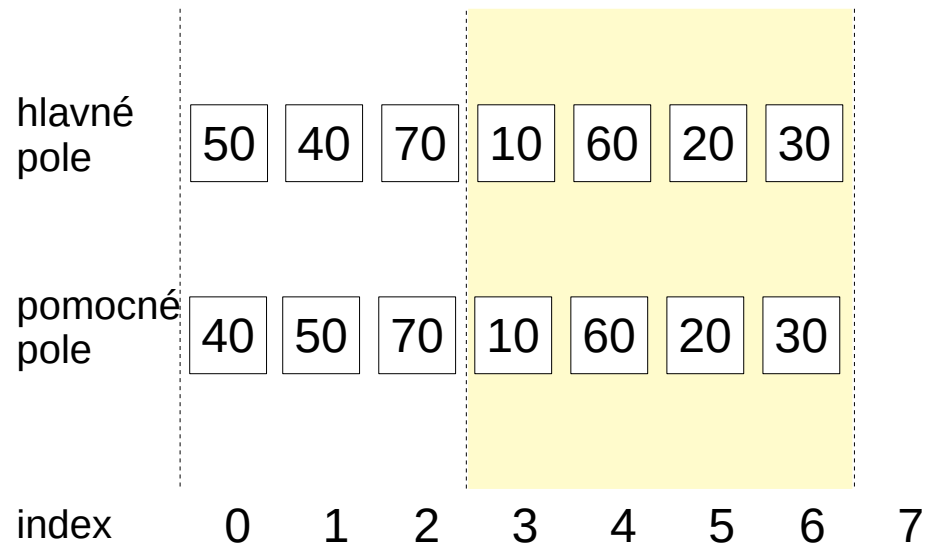
Merge sort – top down



mergesort(0, 7)



Merge sort – top down



mergesort(0, 7)

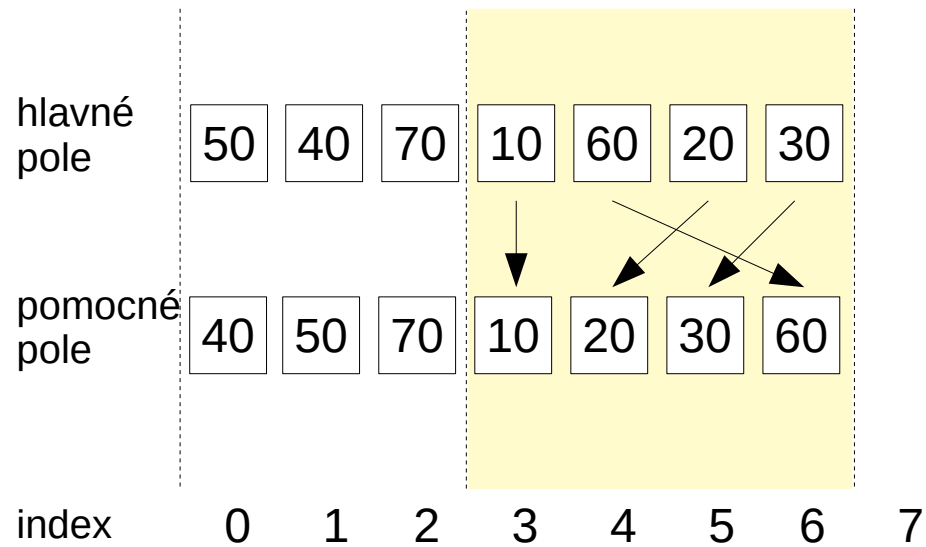
mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), **merge(3, 5, 7)**

↓ copy

↑ copy

Merge sort – top down



mergesort(0, 7)

mergesort(0, 3), **mergesort(3, 7)**, merge(0, 3, 7)

mergesort(3, 5), mergesort(5, 7), **merge(3, 5, 7)**

↓ copy

↑ copy

Merge sort – top down

hlavné pole	50	40	70	10	60	20	30	
pomocné pole	40	50	70	10	20	30	60	
index	0	1	2	3	4	5	6	7

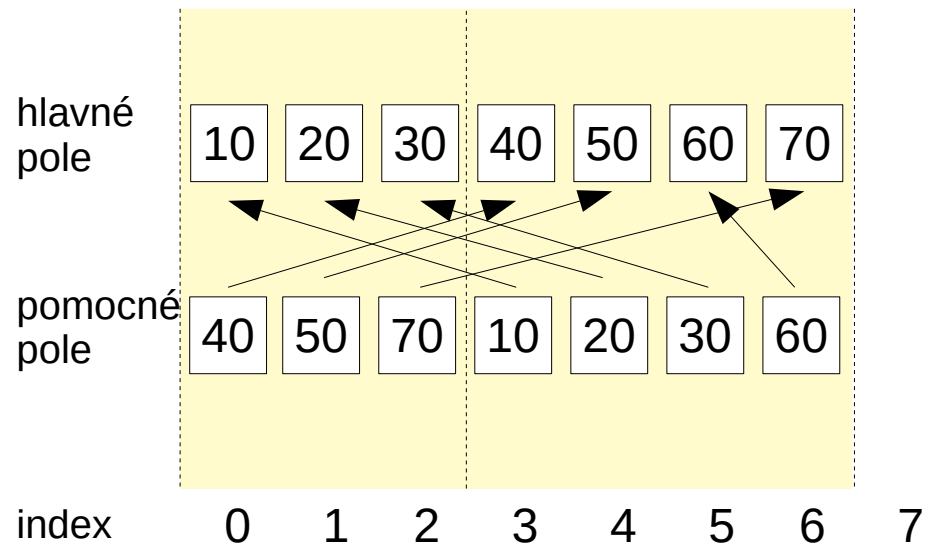
mergesort(0, 7)

mergesort(0, 3), mergesort(3, 7), **merge(0, 3, 7)**

↓
copy

↑
copy

Merge sort – top down



mergesort(0, 7)

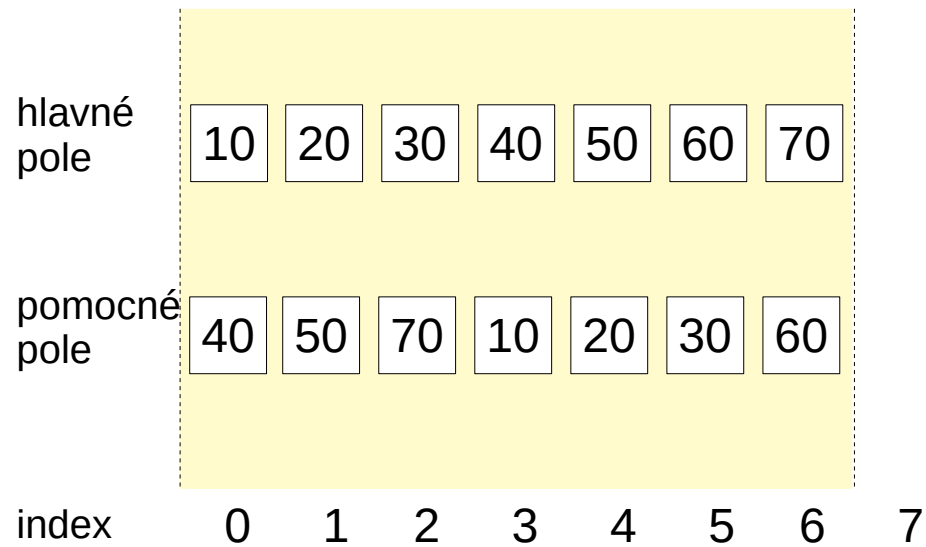
mergesort(0, 3), mergesort(3, 7), **merge(0, 3, 7)**

↑ copy

↓ copy

↑ copy

Merge sort – top down

`mergesort(0, 7)`↑
copy↓
copy↑
copy