

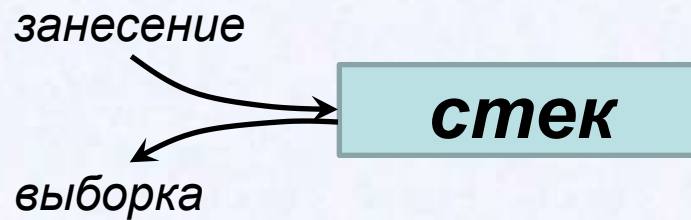
2. Элементарные структуры данных

Элементарные структуры данных

- Стек
- Очередь
- Дек

Элементарные структуры данных

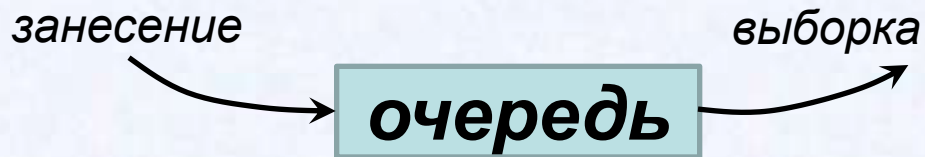
- **Стек**



- вектор
- линейный односвязный список

Элементарные структуры данных

- Очередь



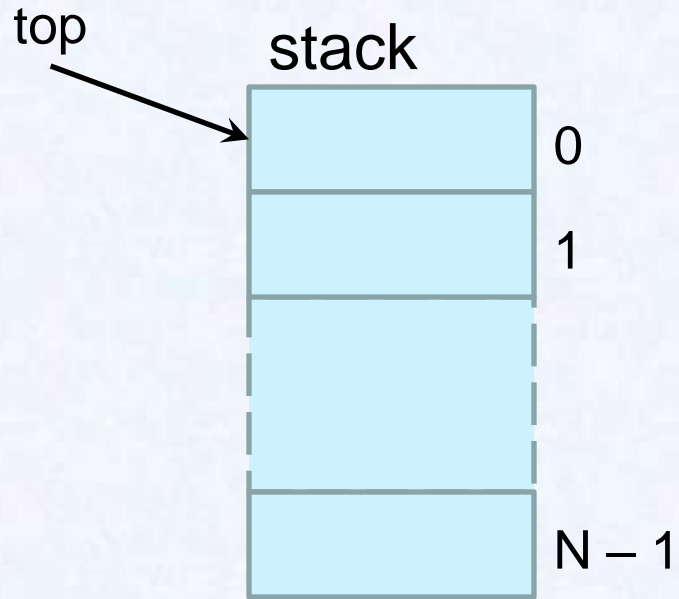
- вектор
- линейный односвязный список
- циклический односвязный список

Элементарные структуры данных

- Дек

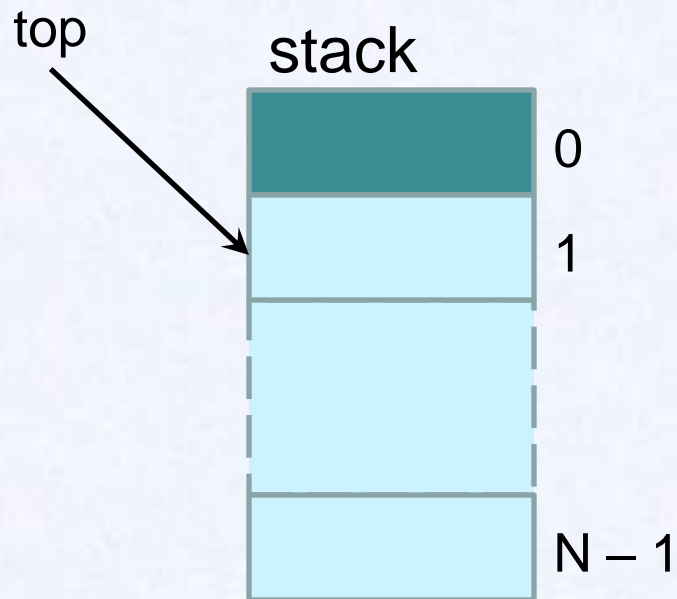


Стек – вектор



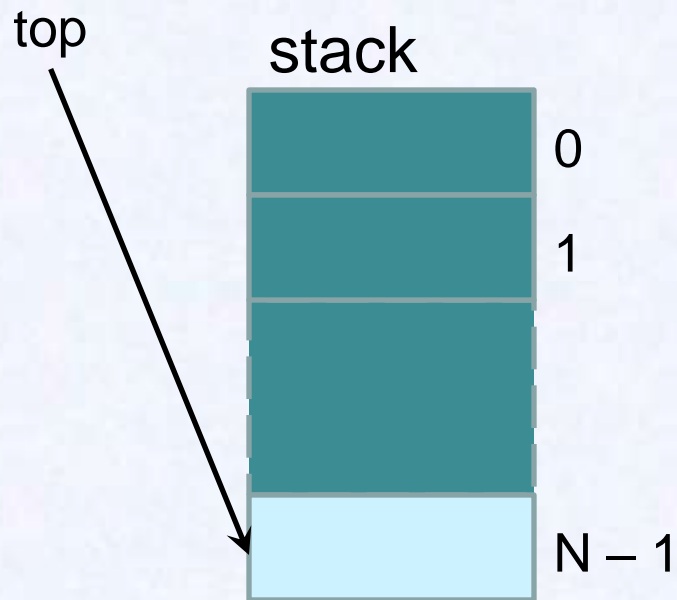
Стек пуст: $\text{top} = 0$

Запись в стек – вектор



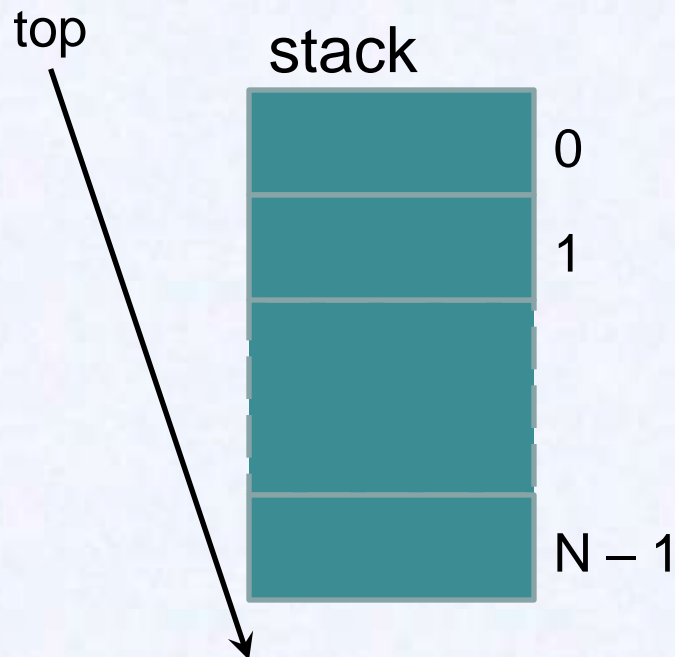
Запись в стек:
`stack[top] = info;`
`top = top + 1;`

Запись в стек – вектор



Запись в стек:
`stack[top] = info;`
`top = top + 1;`

Запись в стек – вектор



Запись в стек:
`stack[top] = info;`
`top = top + 1;`

Стек полон: $\text{top} = N$

Запись в стек – вектор

Запись в стек

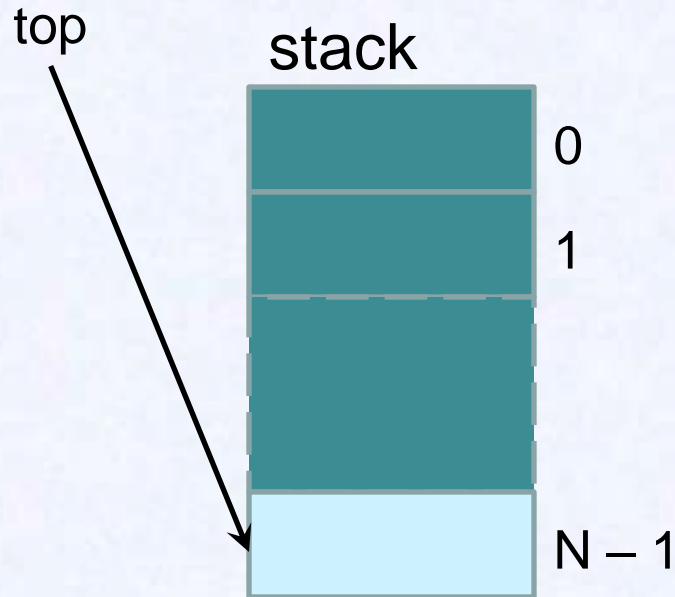


*if стек полон
Отказ*

*Запись в стек по
значению top*

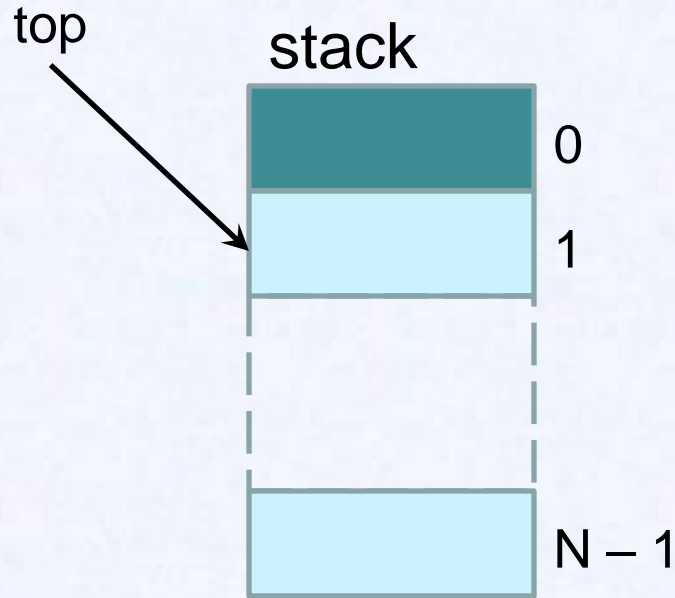
$top = top + 1$
Успех

Чтение из стека – вектора



Чтение из стека:
 $\text{top} = \text{top} - 1;$
 $\text{info} = \text{stack}[\text{top}];$

Чтение из стека – вектора



Чтение из стека:
 $\text{top} = \text{top} - 1;$
 $\text{info} = \text{stack}[\text{top}];$

Чтение из стека – вектора

Чтение из стека



if стек пуст

Отказ

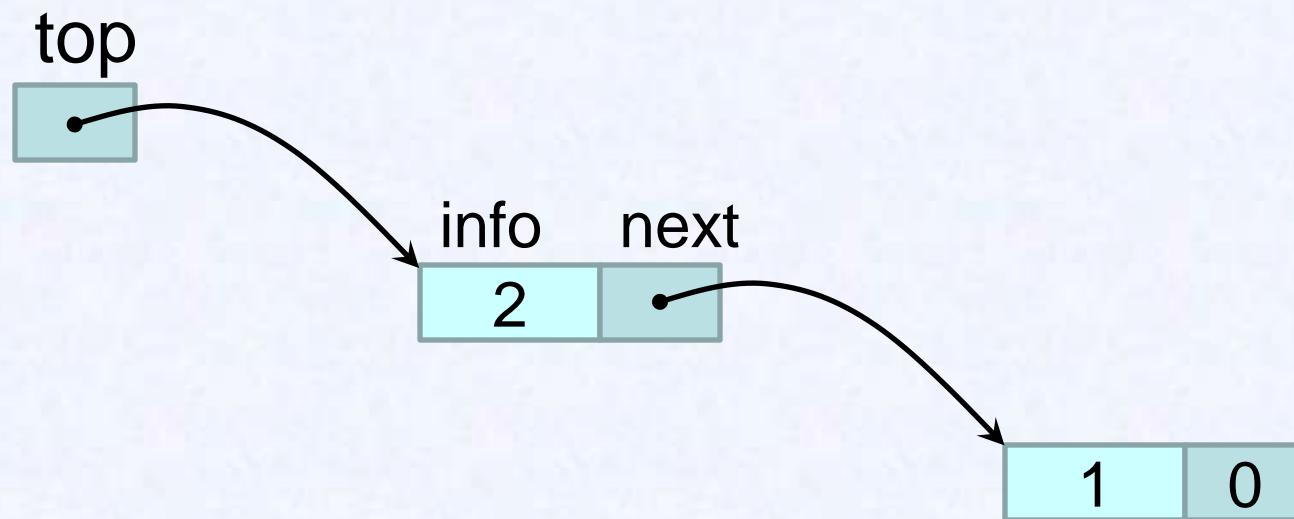
$top = top - 1$

Чтение из стека

по значению top

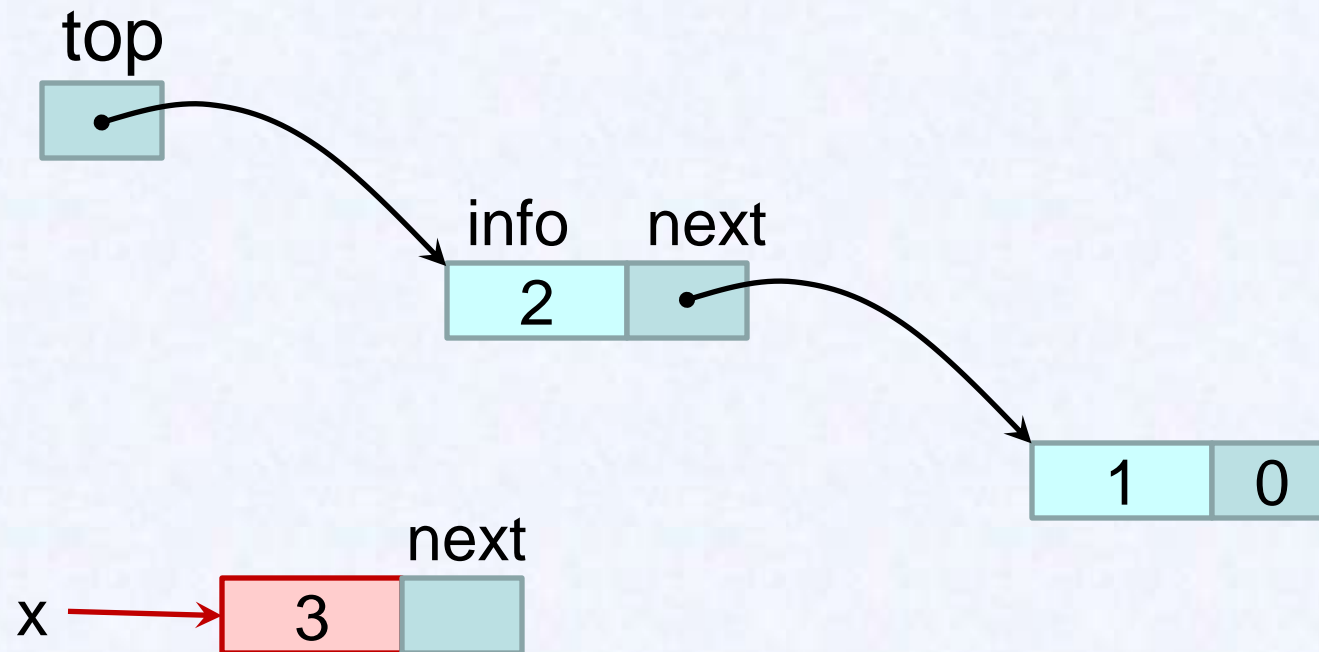
Успех

Стек – список



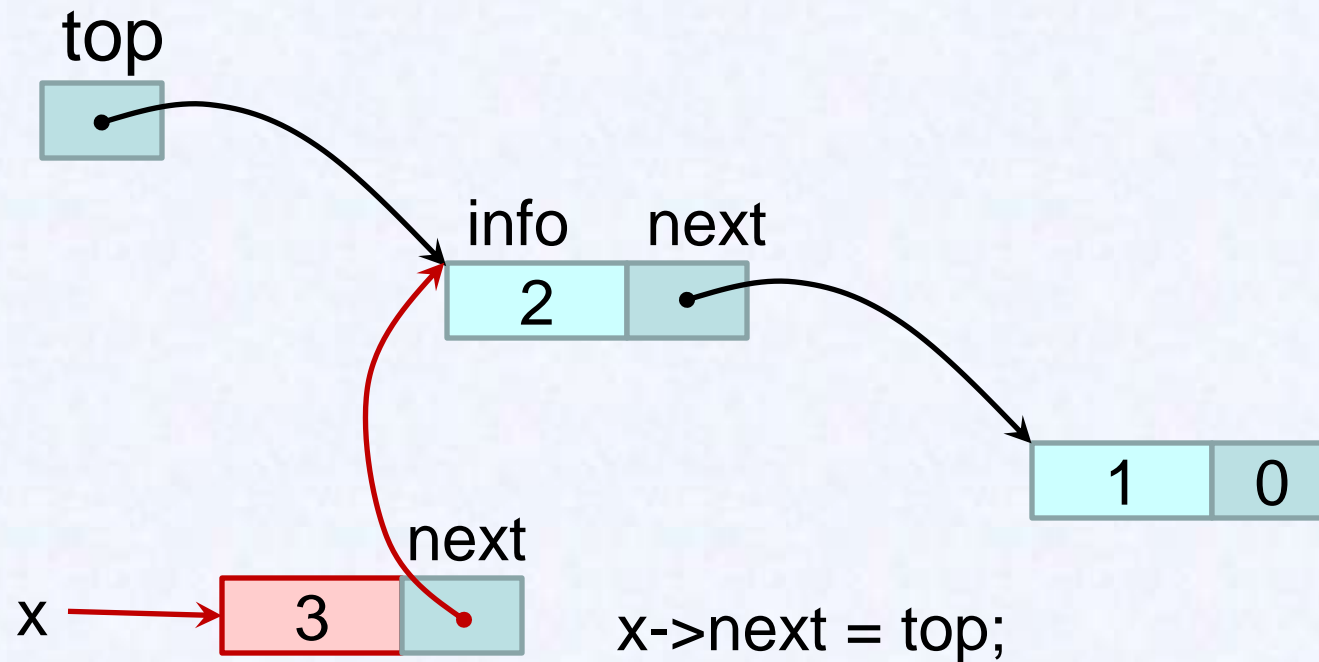
Запись в стек – список

Запись в стек



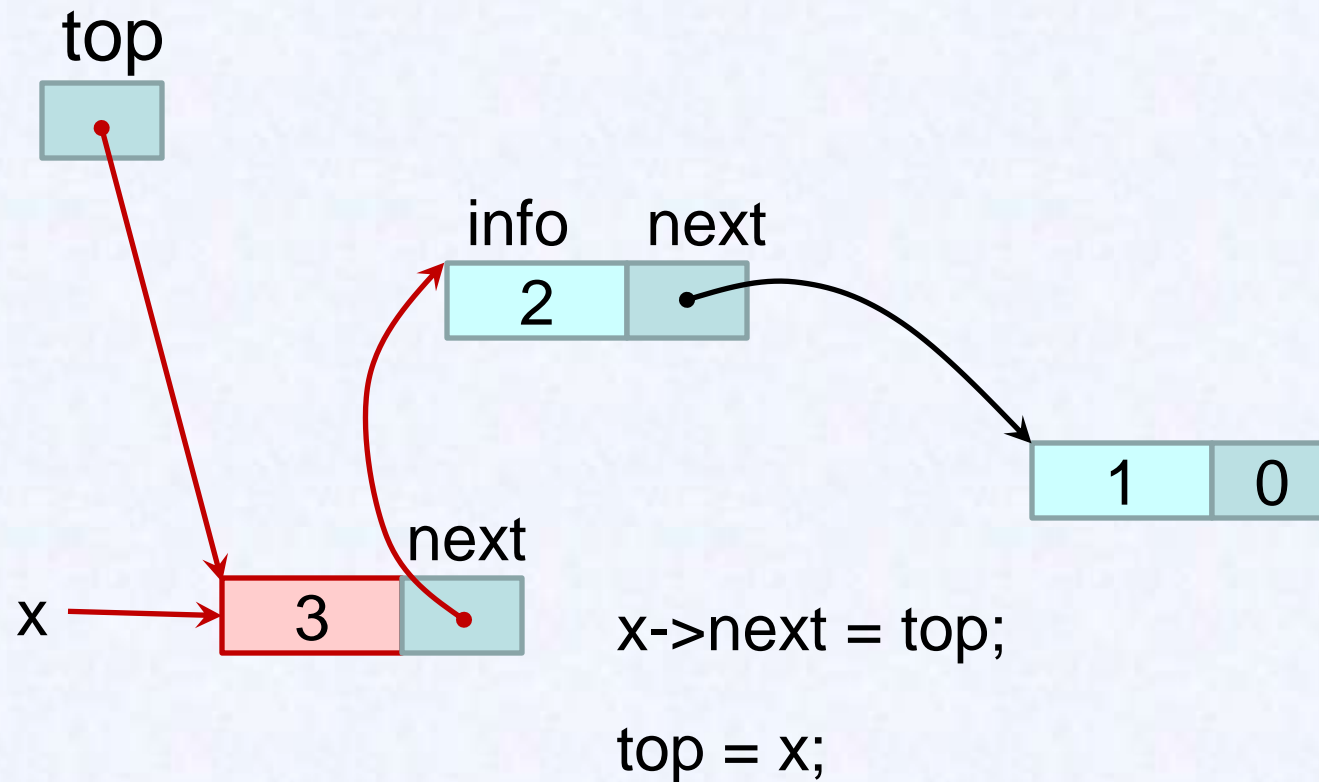
Запись в стек – список

Запись в стек



Запись в стек – список

Запись в стек



Запись в стек – список

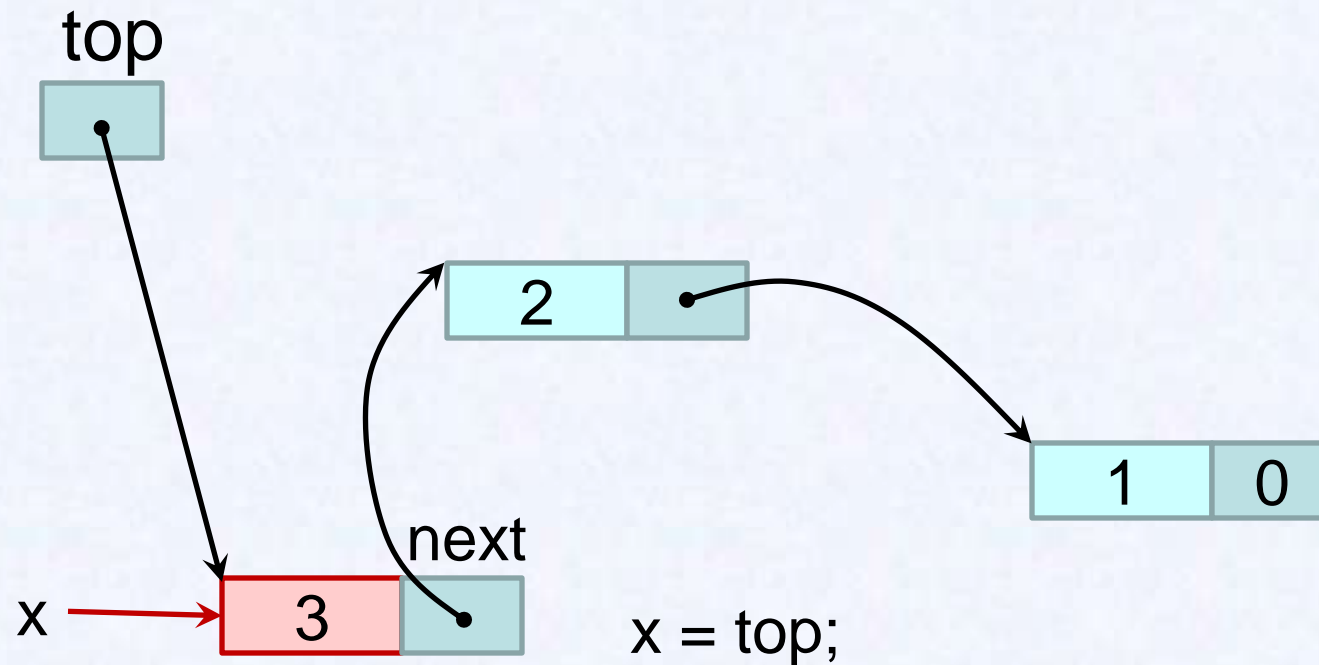
Запись в стек

$x \rightarrow \text{next} = \text{top}$

$\text{top} = x$

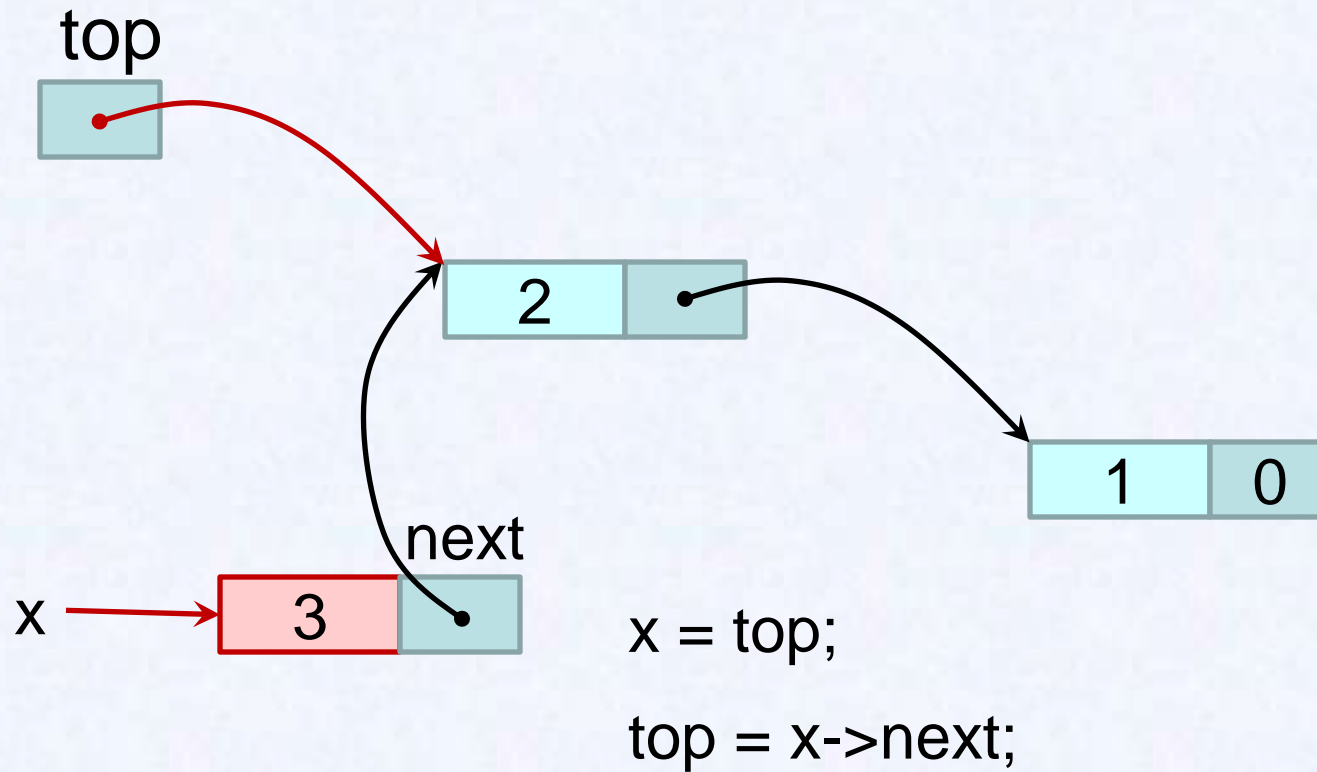
Чтение из стека – списка

Чтение из стека



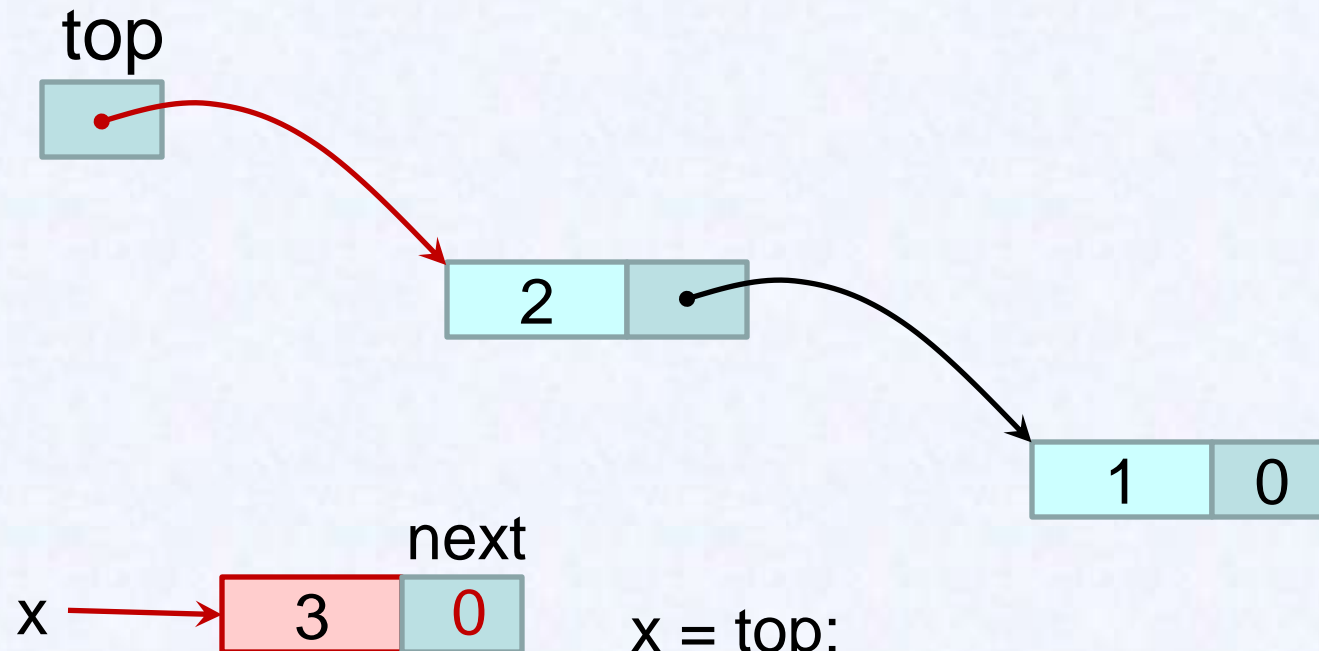
Чтение из стека – списка

Чтение из стека



Чтение из стека – списка

Чтение из стека



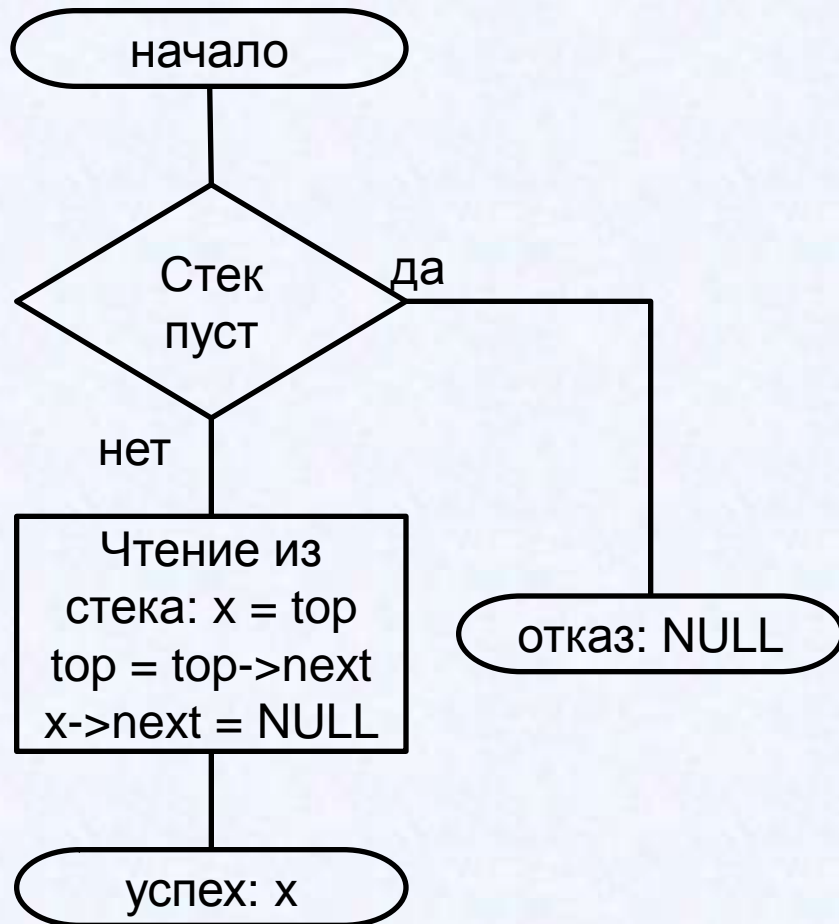
```
x = top;
```

```
top = x->next;
```

```
x->next = 0;
```

Чтение из стека – списка

Чтение из стека



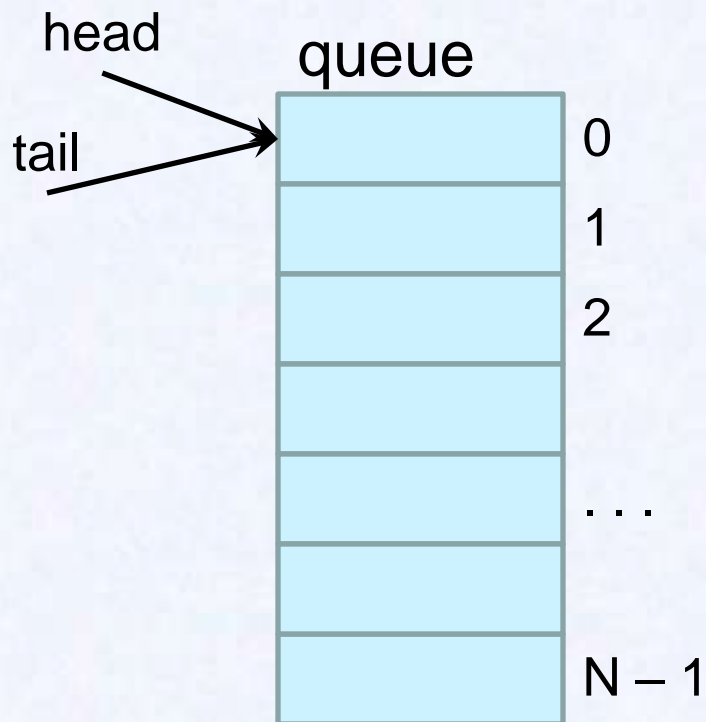
if *стек пуст*
отказ: $x = \text{NULL}$

$x = \text{top}$

$\text{top} = \text{top} \rightarrow \text{next}$

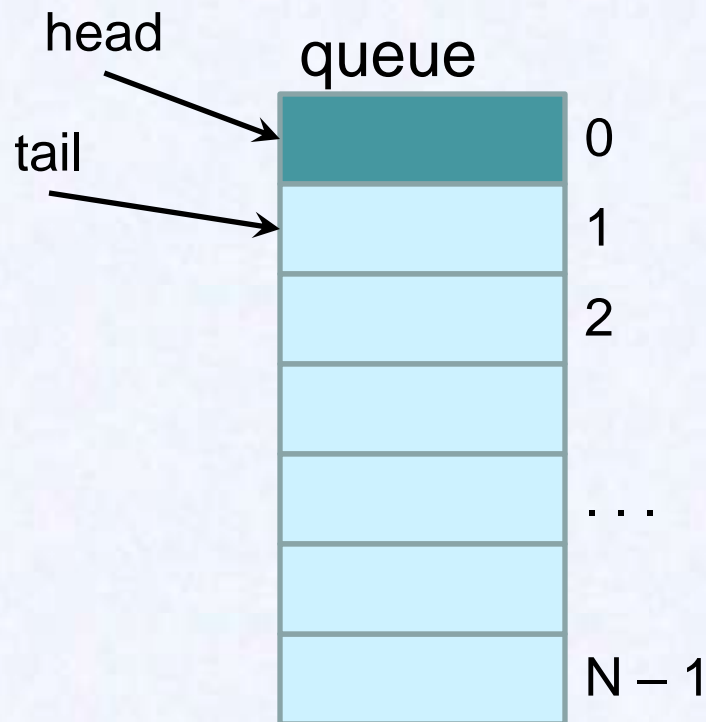
$x \rightarrow \text{next} = \text{NULL}$

Очередь – вектор



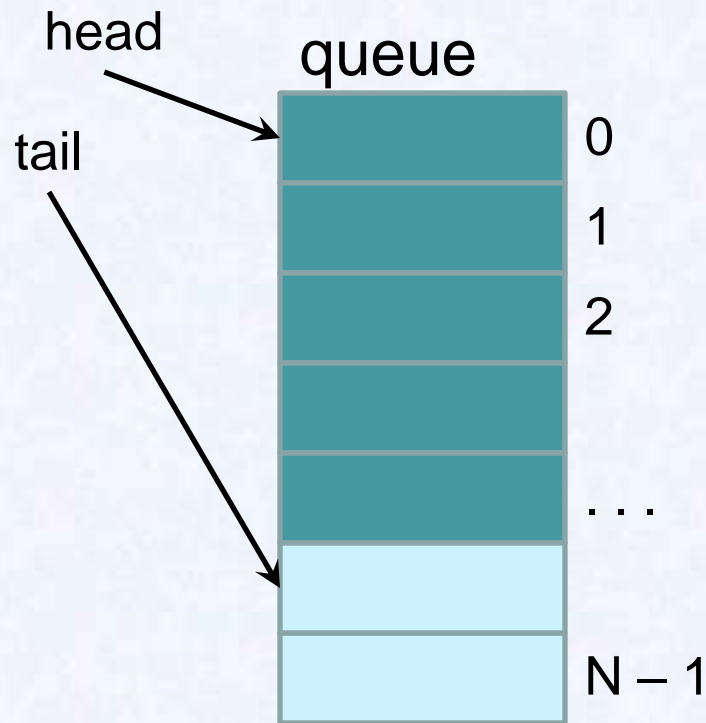
Очередь пуста: **head == tail**

Запись в очередь – вектор



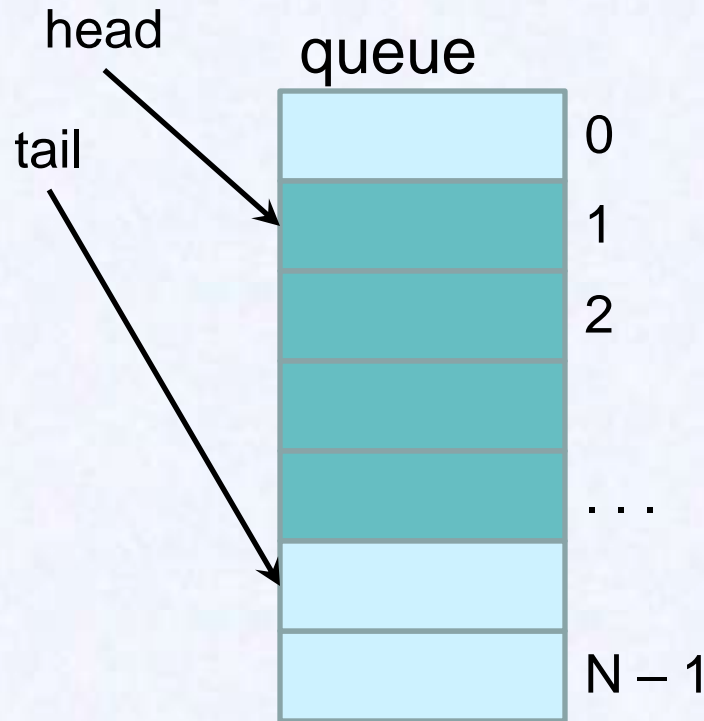
Запись в очередь:
`queue[tail] = info;`
`tail = tail + 1;`

Запись в очередь – вектор



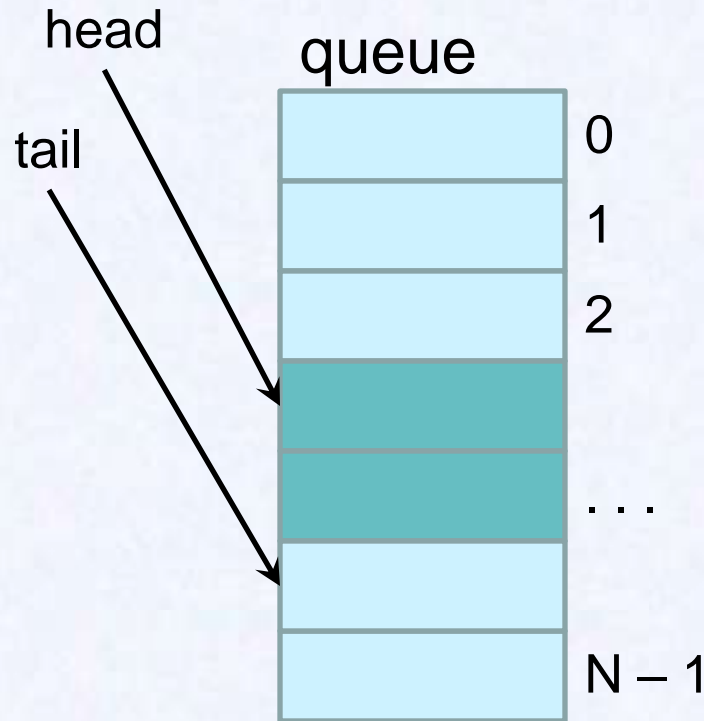
Запись в очередь:
 $\text{queue}[\text{tail}] = \text{info};$
 $\text{tail} = \text{tail} + 1;$

Запись в очередь – вектор



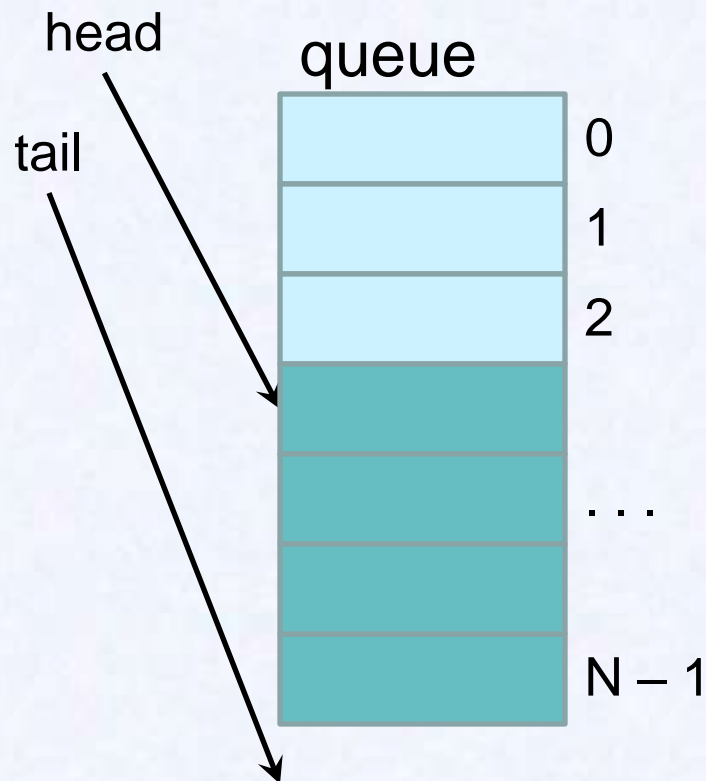
Чтение из очереди:
`info = queue[head];`
`head = head + 1;`

Запись в очередь – вектор



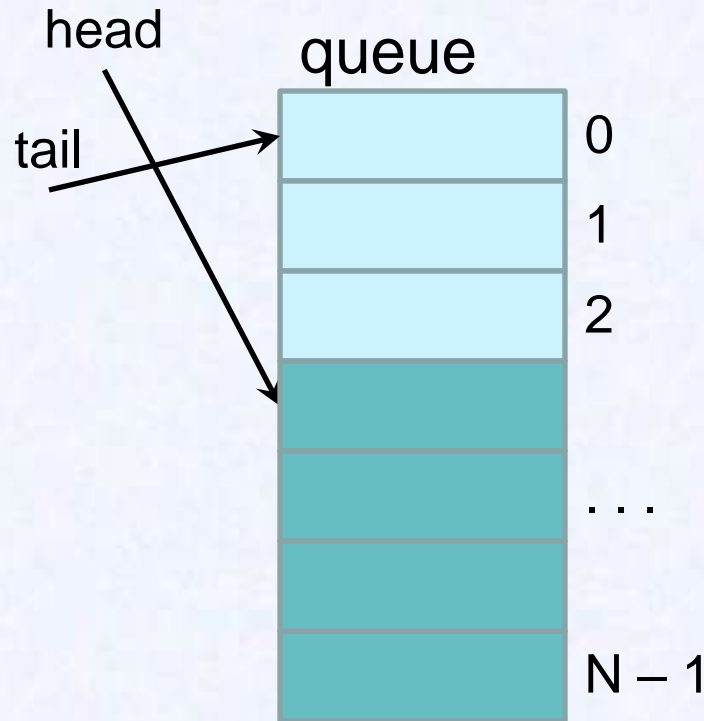
Чтение из очереди:
`info = queue[head];`
`head = head + 1;`

Запись в очередь – вектор



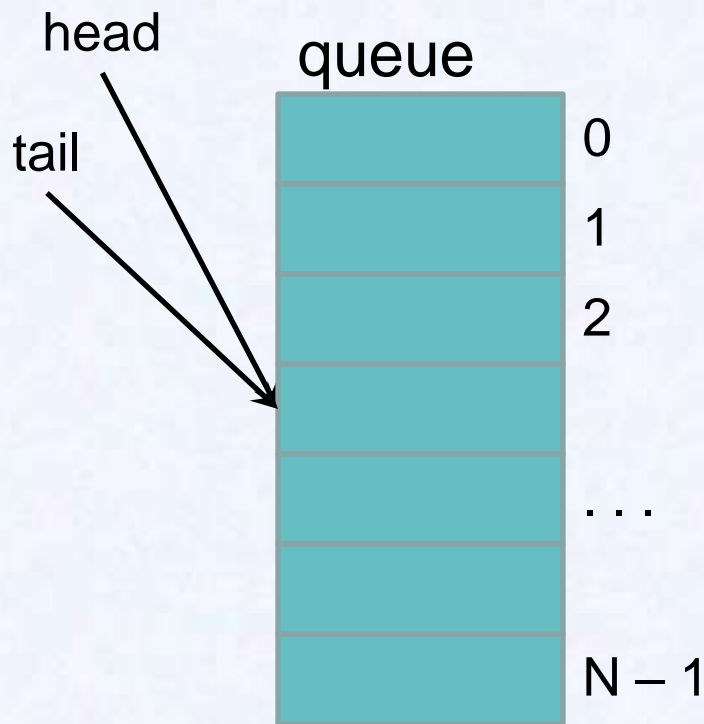
Запись в очередь:
`queue[tail] = info;`

Запись в очередь – вектор



Запись в очередь:
 $\text{queue}[\text{tail}] = \text{info};$
 $\text{tail} = (\text{tail} + 1) \% N;$

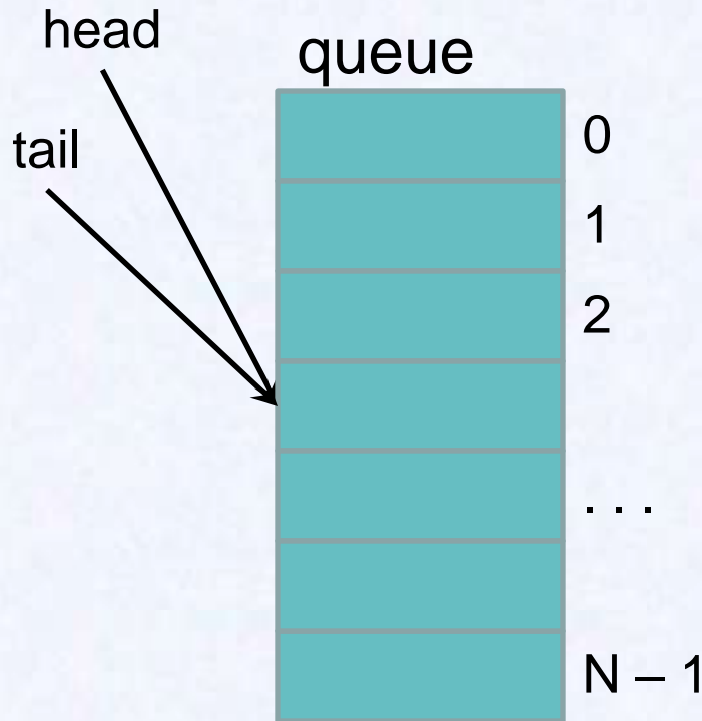
Запись в очередь – вектор



Запись в очередь:
`queue[tail] = info;`
`tail = (tail + 1) % N;`

Очередь полна: **head == tail**

Запись в очередь – вектор



Запись в очередь:
 $\text{queue}[\text{tail}] = \text{info};$
 $\text{tail} = (\text{tail} + 1) \% N;$
 $n = n + 1;$

Очередь полна: $\text{head} == \text{tail}$

$n == N$

Запись в очередь – вектор

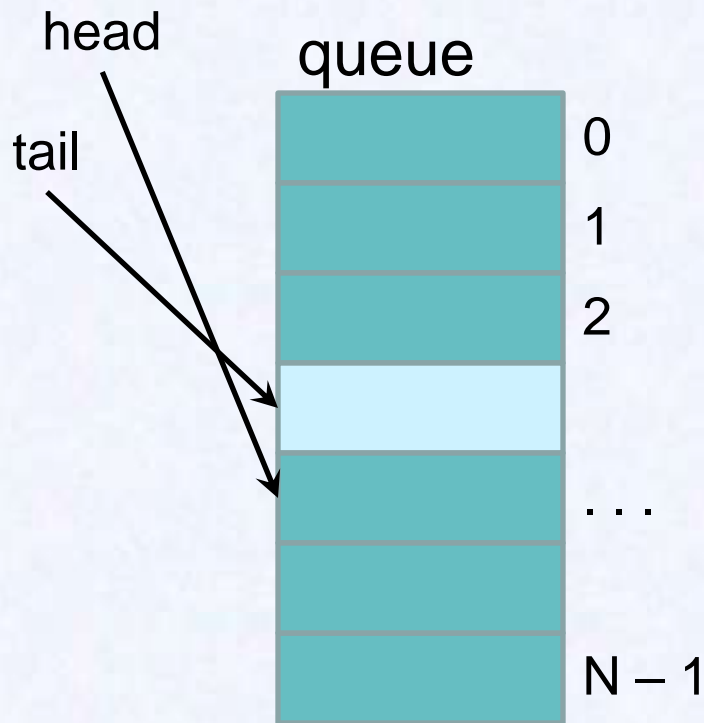
Запись в очередь



*if очередь полна
Отказ*

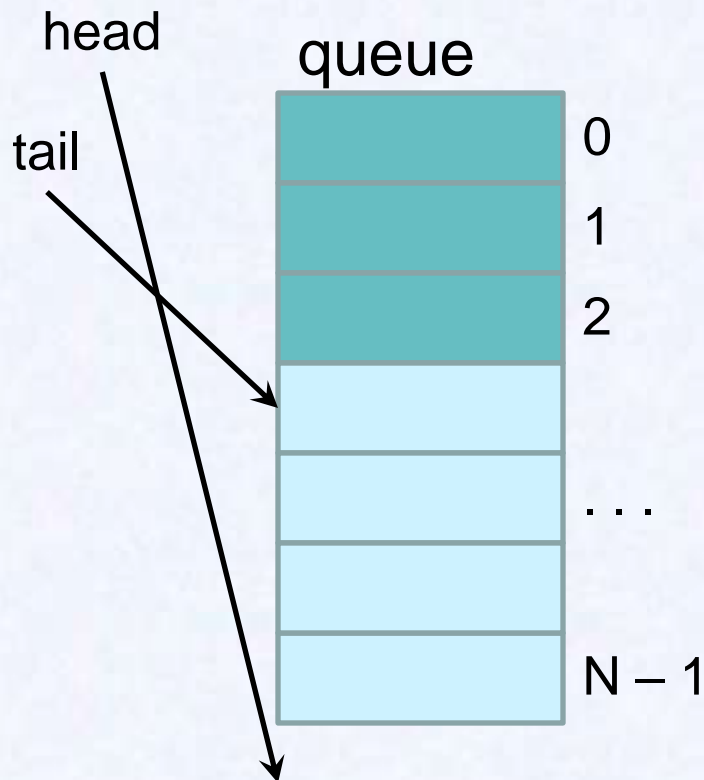
*Запись в очередь
по значению tail
 $tail = (tail + 1) \% N$*

Чтение из очереди – вектора



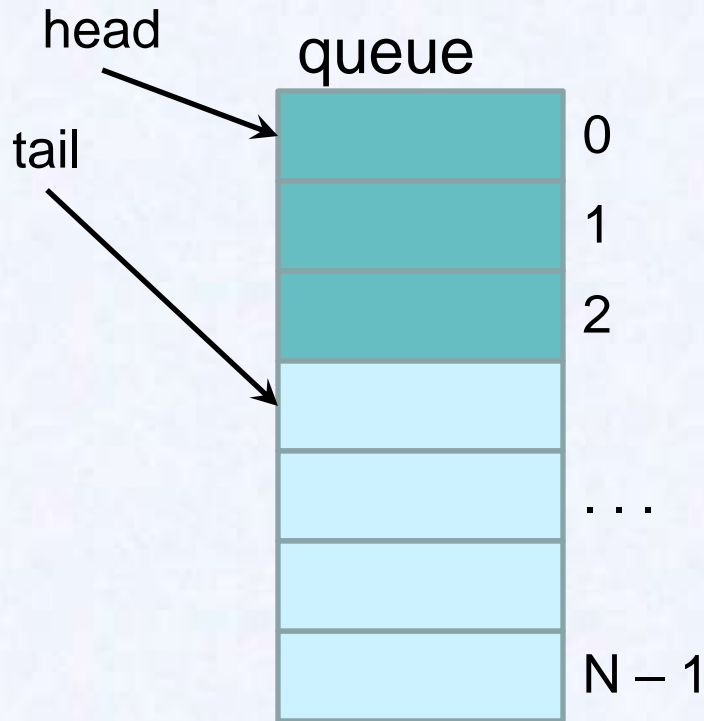
Чтение из очереди:
`info = queue[head];`
`head = head + 1;`

Чтение из очереди – вектора



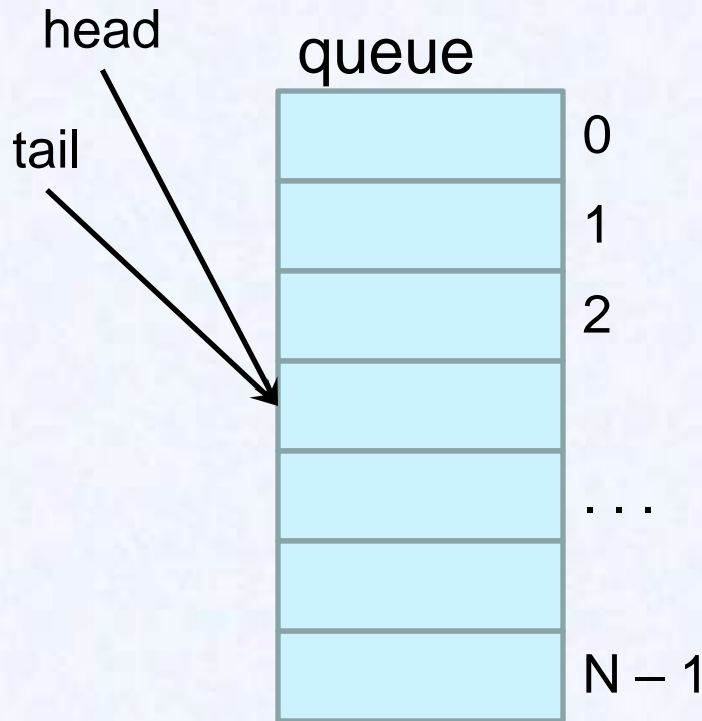
Чтение из очереди:
`info = queue[head];`
`head = (head + 1) % N;`

Чтение из очереди – вектора



Чтение из очереди:
`info = queue[head];`
`head = (head + 1) % N;`

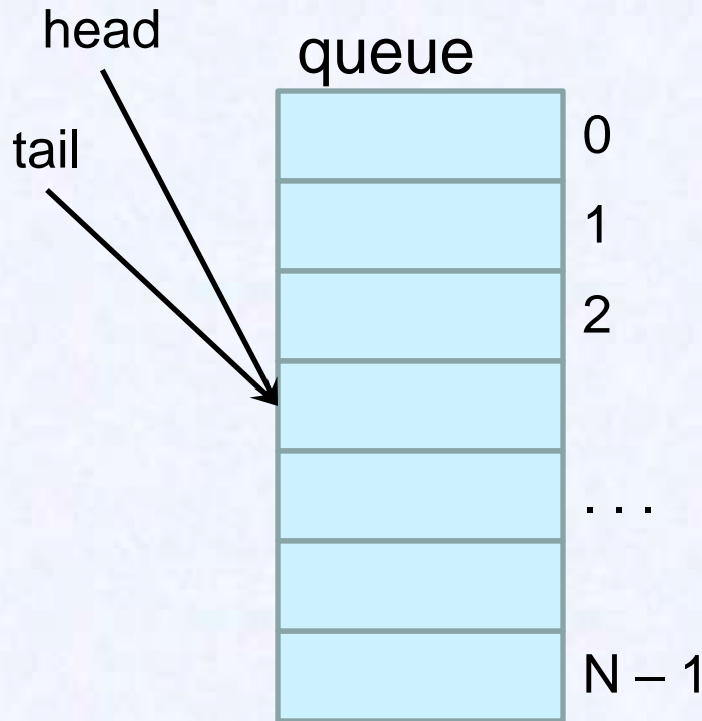
Чтение из очереди – вектора



Чтение из очереди:
`info = queue[head];`
`head = (head + 1) % N;`

Очередь пуста: **head == tail**

Чтение из очереди – вектора



Чтение из очереди:
`info = queue[head];`
`head = (head + 1) % N;`
`n = n - 1;`

Очередь пуста: `head == tail`

`n == 0`

Чтение из очереди – вектора

Чтение из очереди

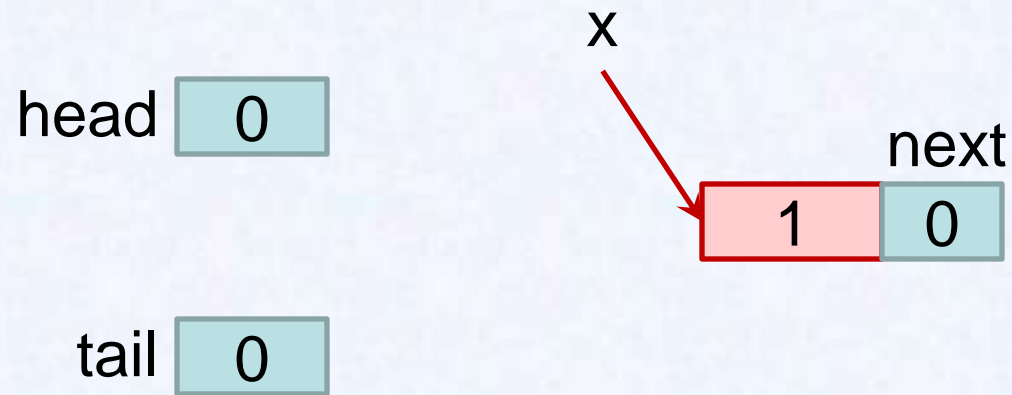


*if очередь пуста
Отказ*

*Чтение из очереди
по значению head
 $head = (head + 1) \% N$*

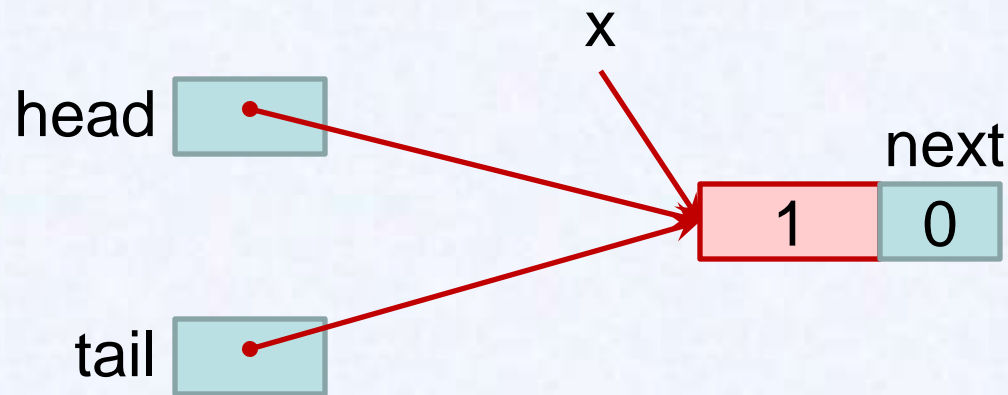
Очередь – линейный список

Очередь пуста: $\text{head} = 0$, $\text{tail} = 0$



Запись в очередь – линейный список

Очередь пуста: $\text{head} = 0$, $\text{tail} = 0$

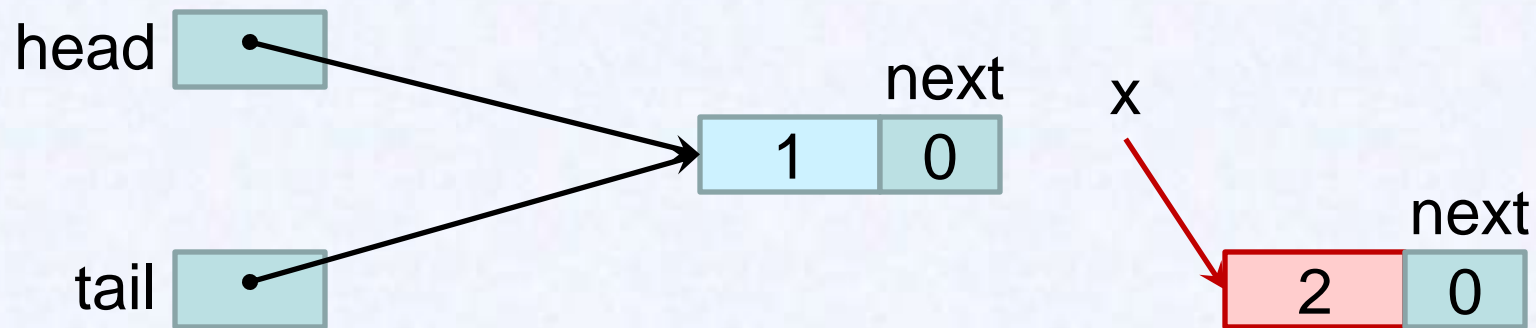


Запись в пустую очередь:

$\text{head} = x$;

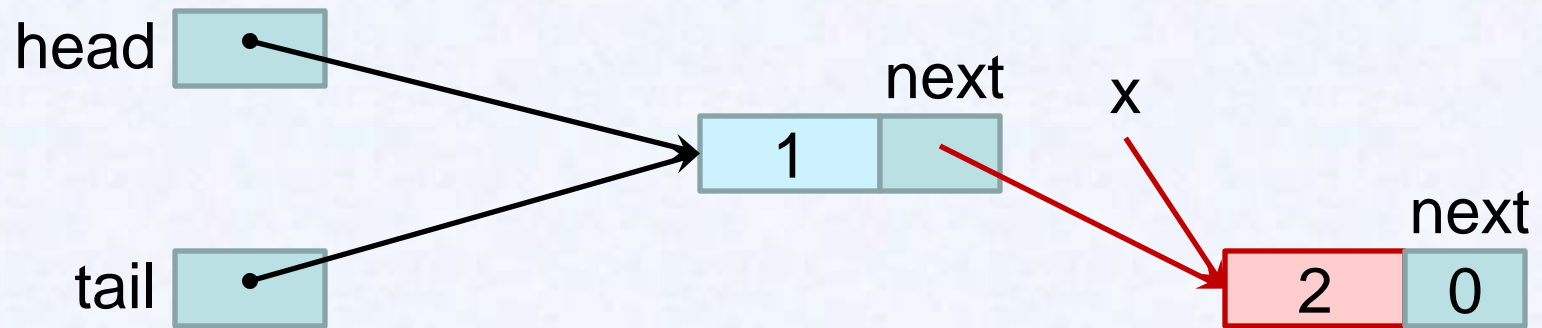
$\text{tail} = x$;

Запись в очередь – линейный список



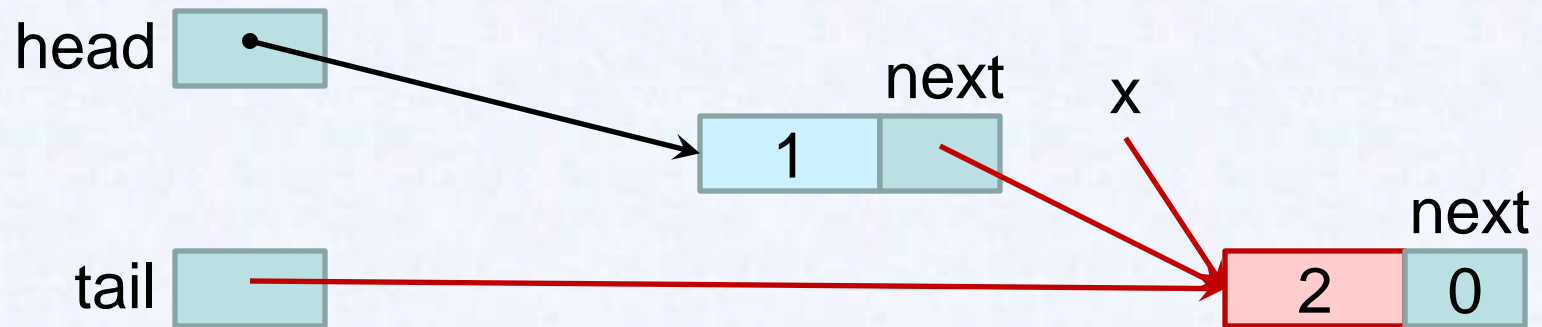
Запись в очередь:

Запись в очередь – линейный список



Запись в очередь:
 $\text{tail} \rightarrow \text{next} = x;$

Запись в очередь – линейный список

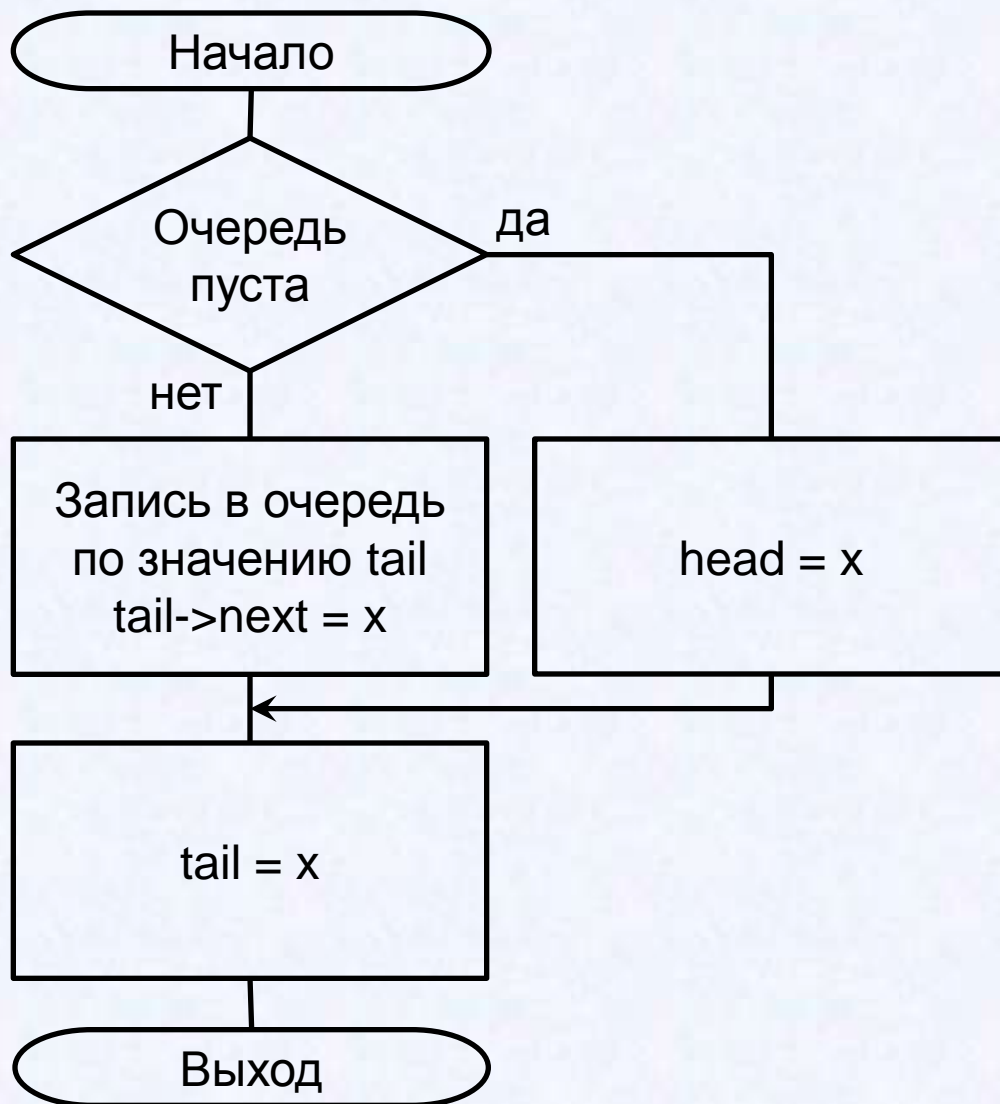


Запись в очередь:

`tail->next = x;`

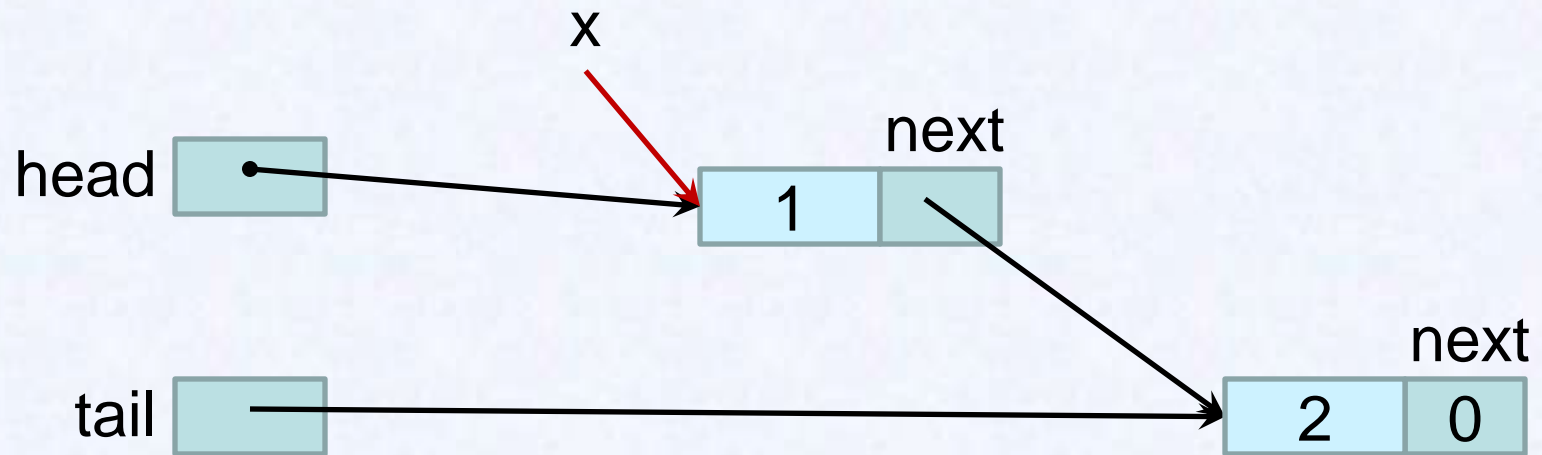
`tail = x;`

Запись в очередь – линейный список



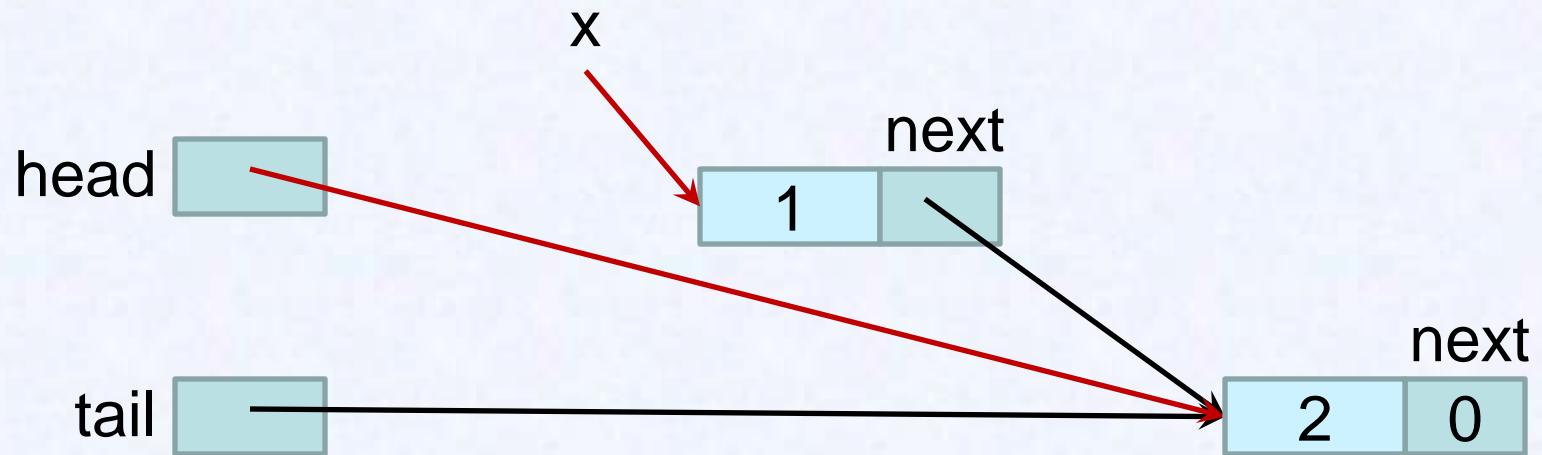
if *очередь пуста*
 $head = x$
else
 $tail \rightarrow next = x$
 $tail = x$

Чтение из очереди – линейного списка



Чтение из очереди:
`x = head;`

Чтение из очереди – линейного списка

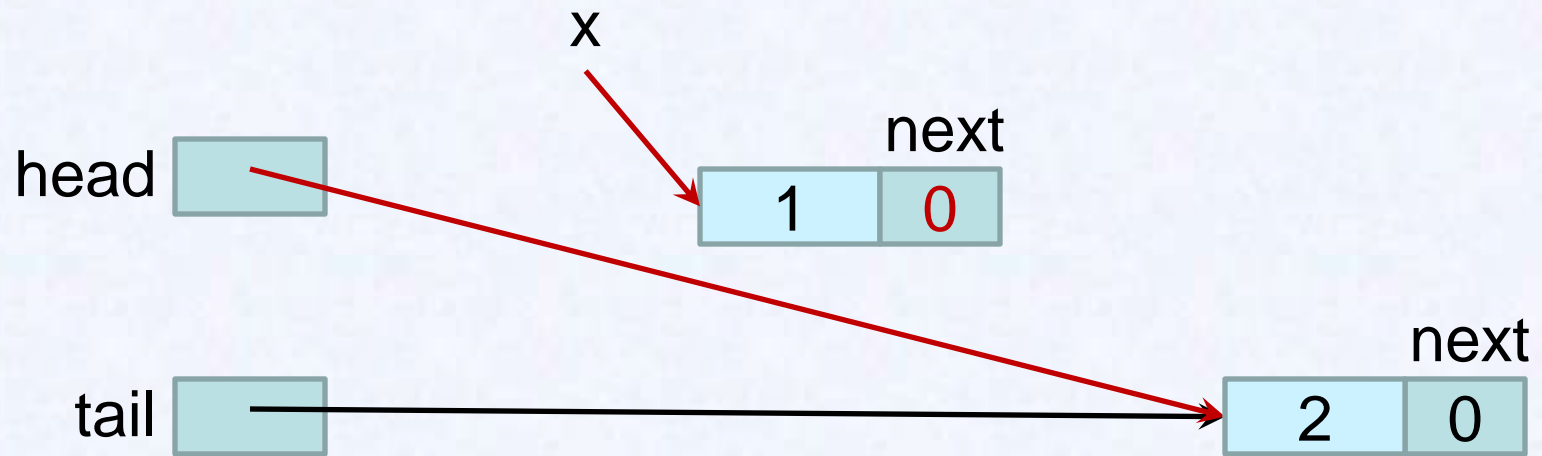


Чтение из очереди:

```
x = head;
```

```
head = head->next;
```

Чтение из очереди – линейного списка



Чтение из очереди:

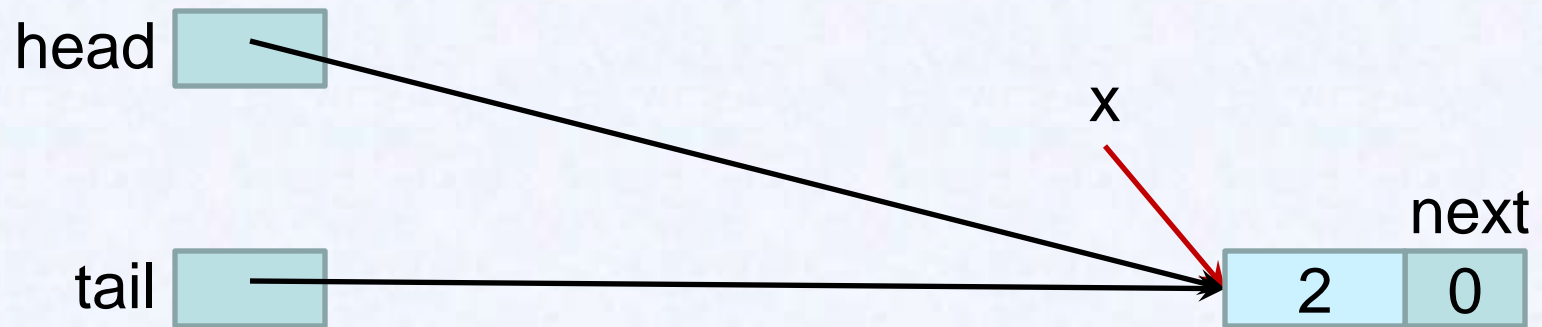
```
x = head;
```

```
head = head->next;
```

```
x->next = 0;
```

2.47

Чтение из очереди – линейного списка



Чтение из очереди последнего элемента:
`x = head;`

Чтение из очереди – линейного списка



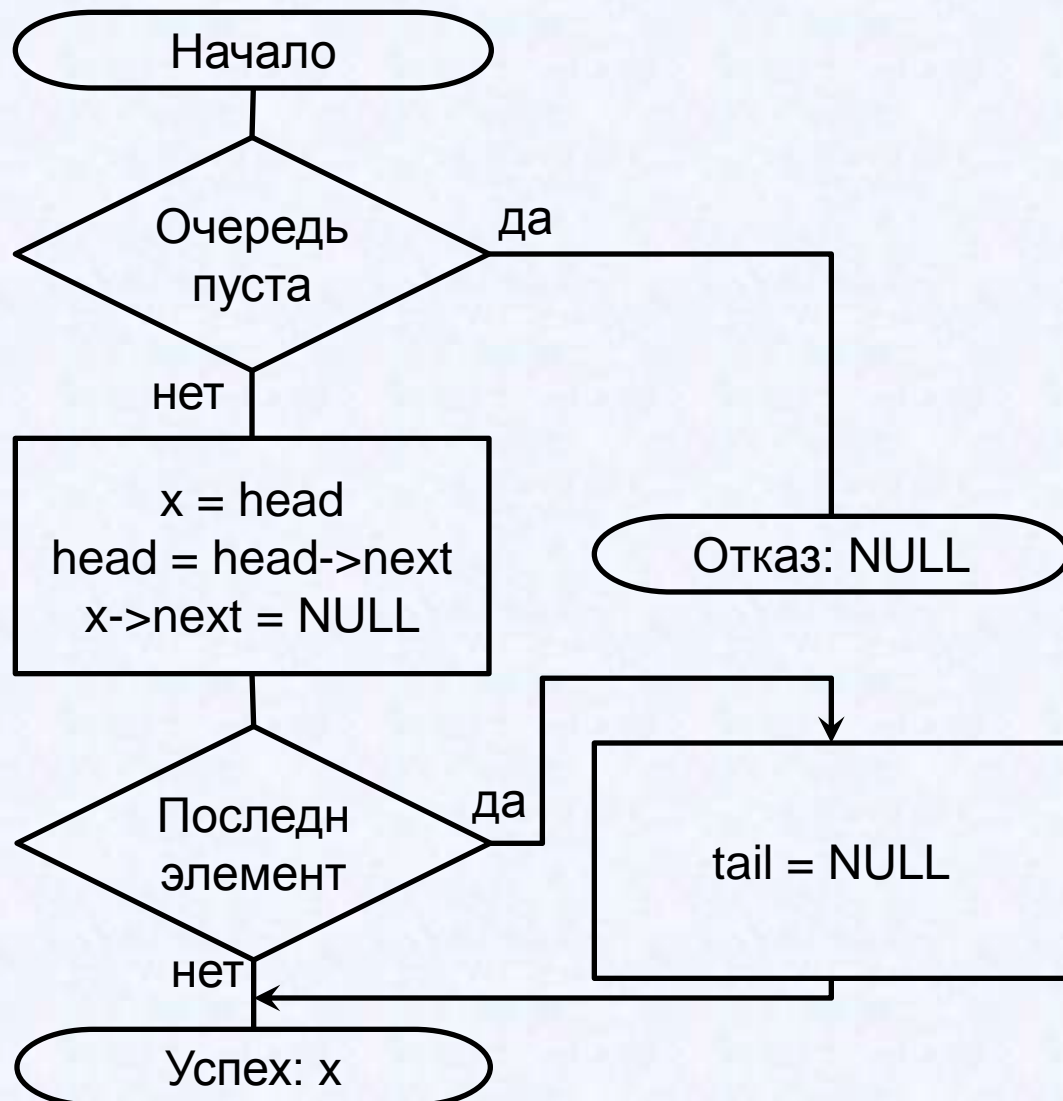
Чтение из очереди последнего элемента:

```
x = head;
```

```
head = 0;
```

```
tail = 0;
```

Чтение из очереди – линейного списка



if очередь пуста
Отказ

`x = head`

`head = head->next`

`x->next = NULL`

if удален последний элемент
`tail = NULL`

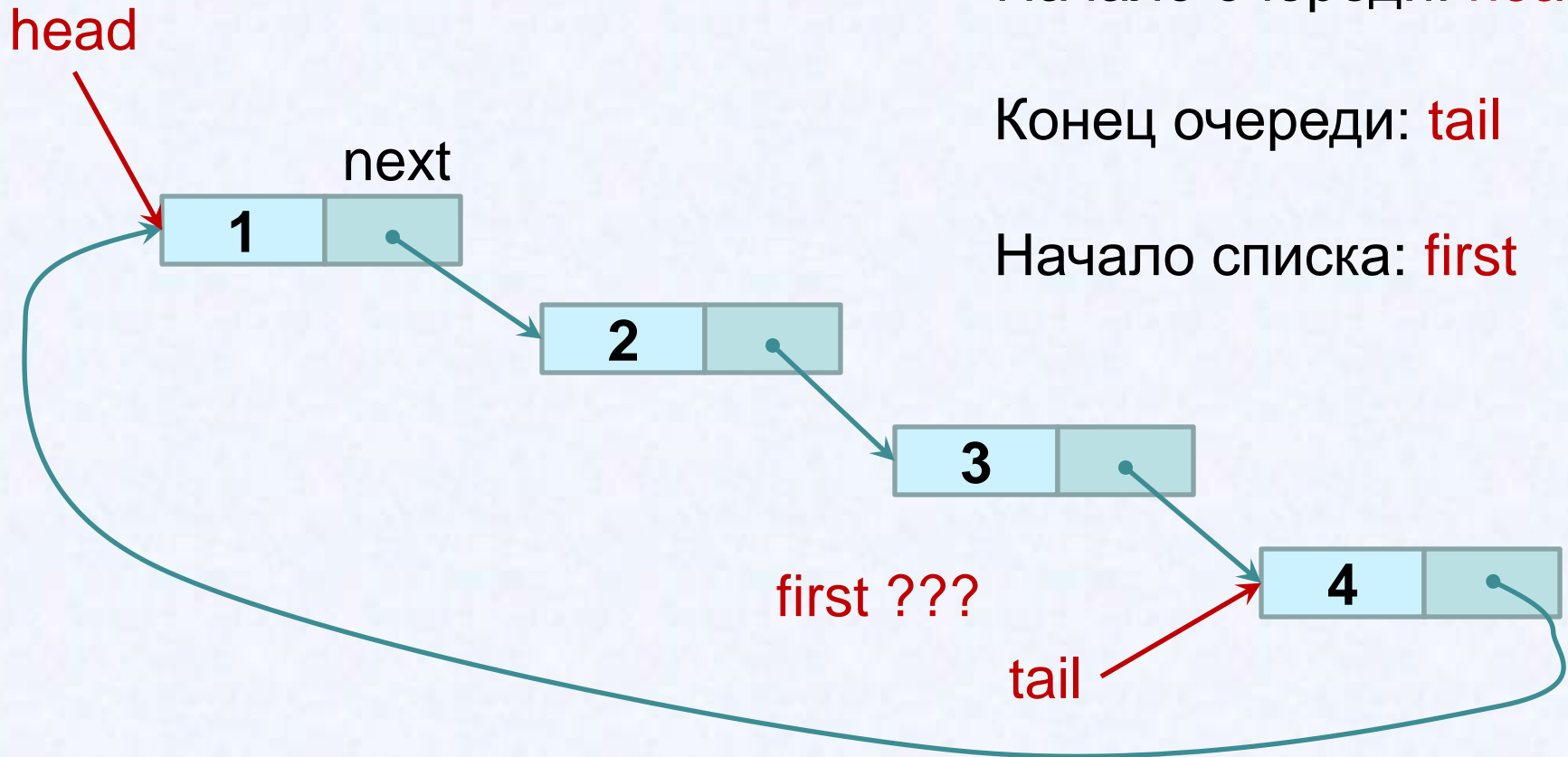
`Успех: x`

Очередь – ЦИКЛИЧЕСКИЙ СПИСОК

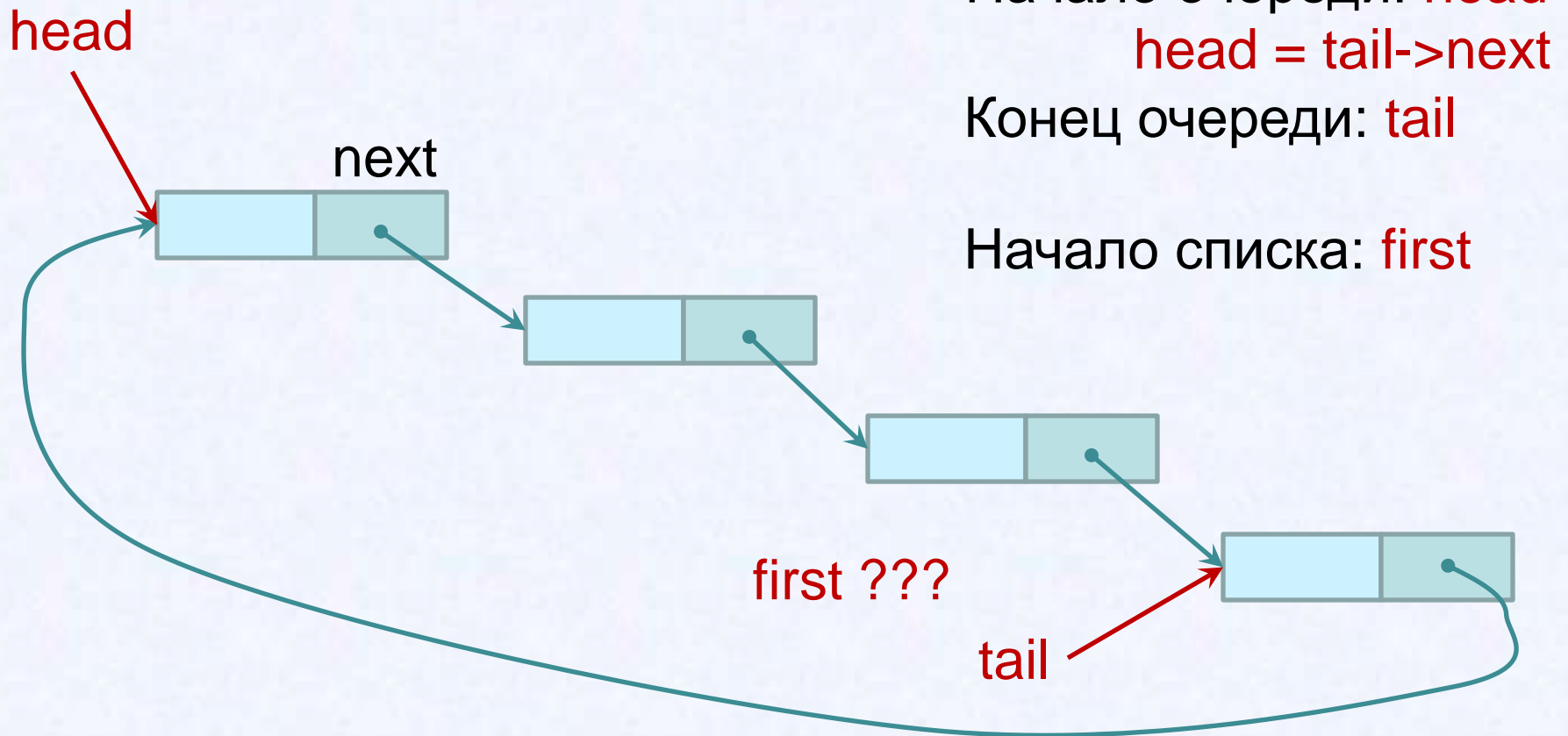
Начало очереди: **head**

Конец очереди: **tail**

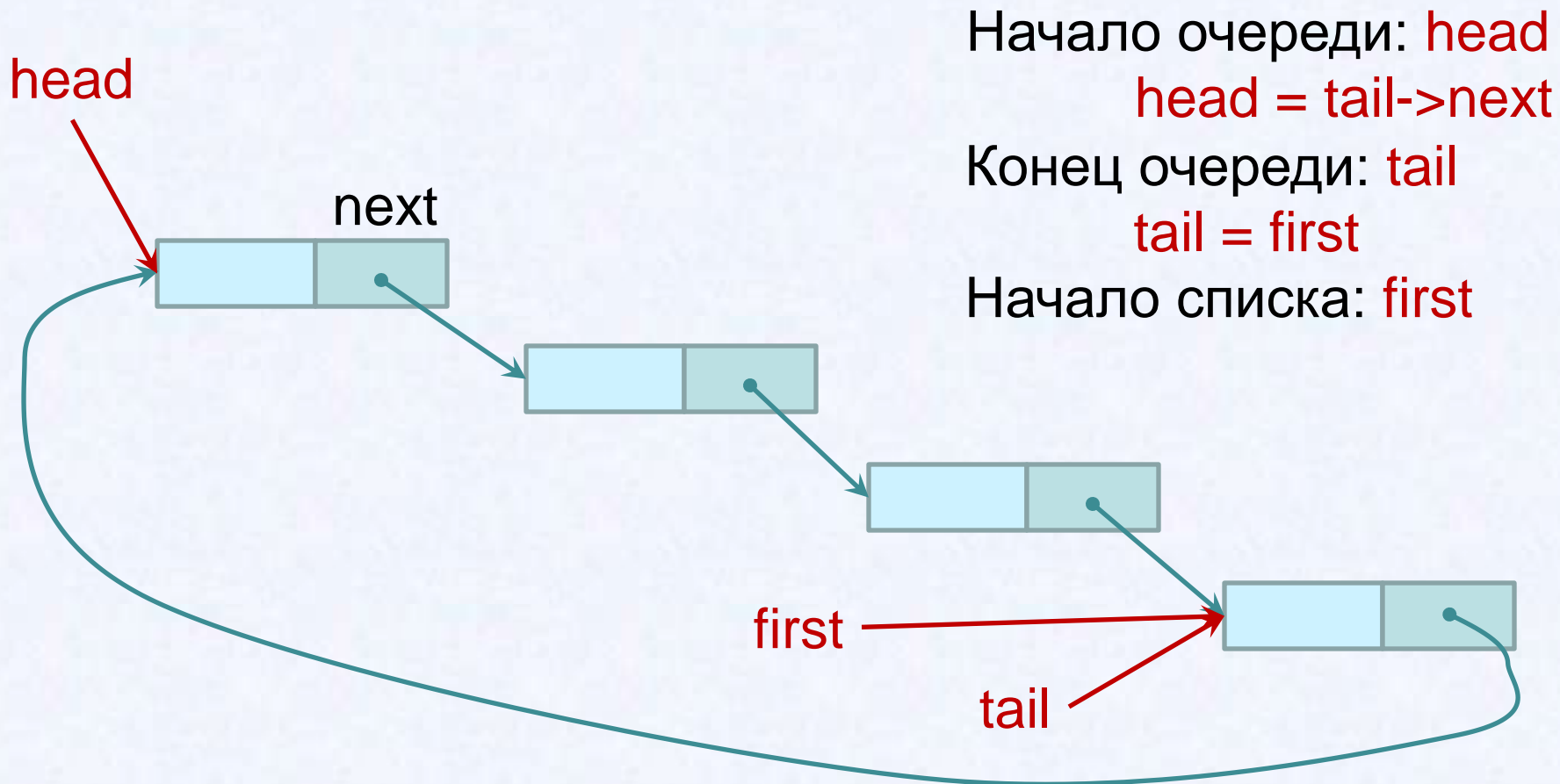
Начало списка: **first**



Очередь – ЦИКЛИЧЕСКИЙ СПИСОК



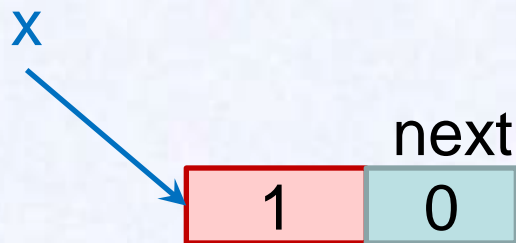
Очередь – ЦИКЛИЧЕСКИЙ СПИСОК



2.53

Запись в очередь – циклический список

Запись в очередь



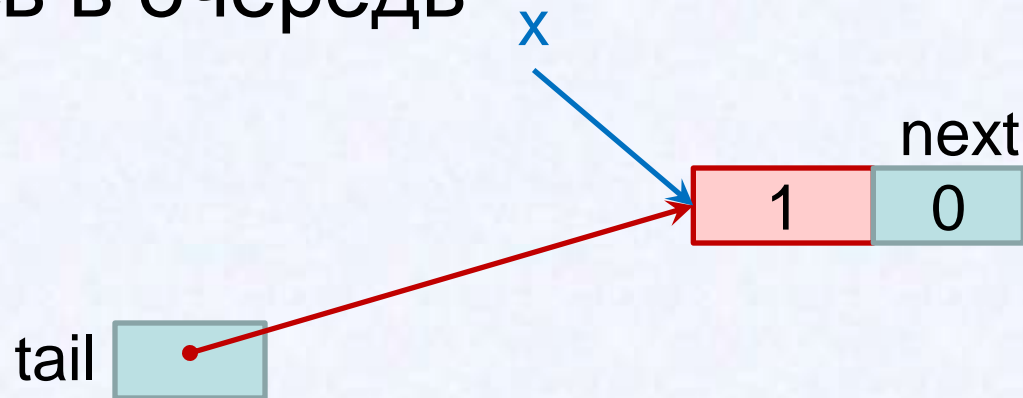
tail

0

Очередь пуста

Запись в очередь – циклический список

Запись в очередь

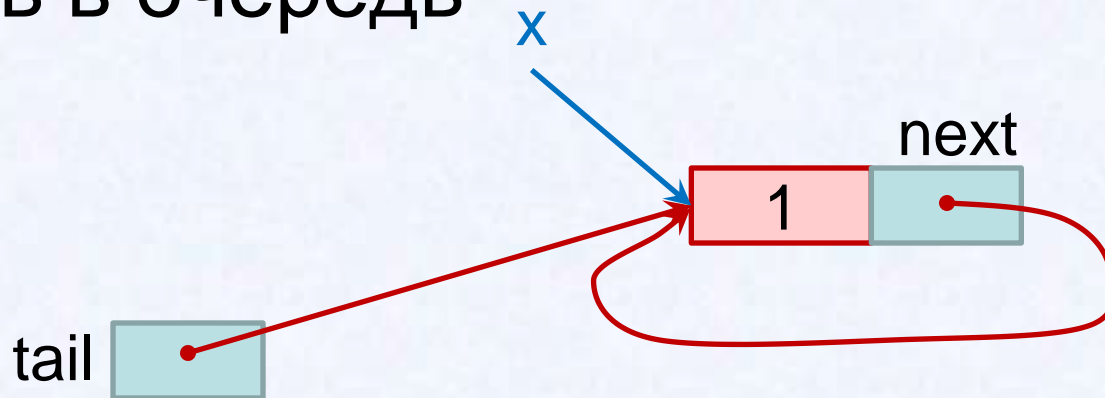


Очередь пуста

$\text{tail} = x$

Запись в очередь – циклический список

Запись в очередь



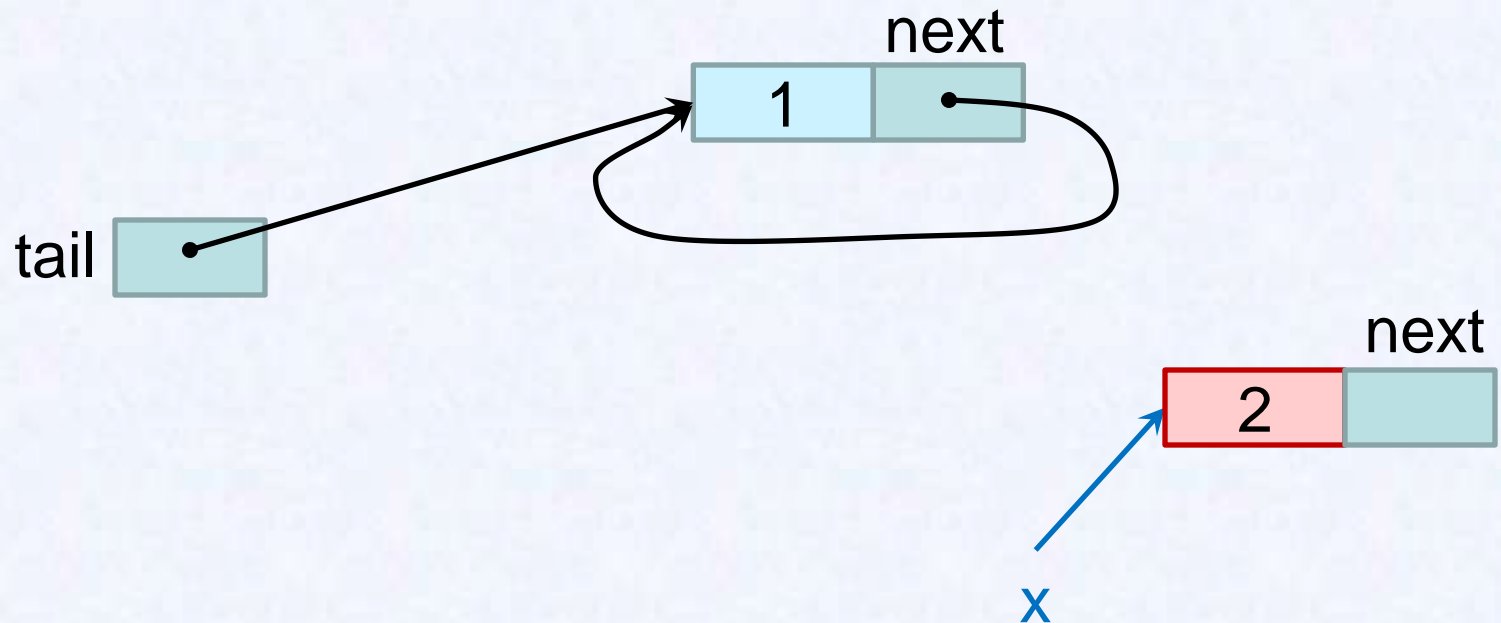
Очередь пуста

$\text{tail} = x$

$\text{tail} \rightarrow \text{next} = x$

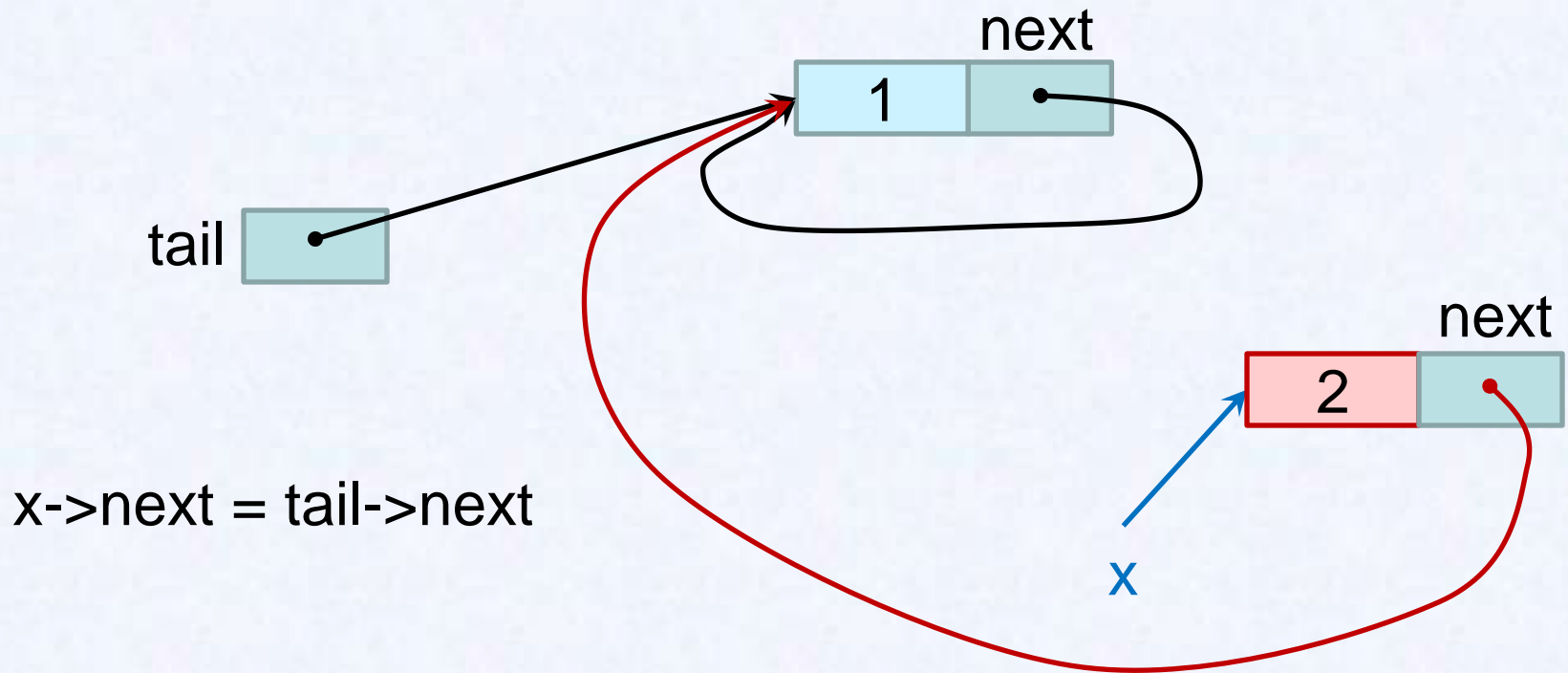
Запись в очередь – циклический список

Запись в очередь



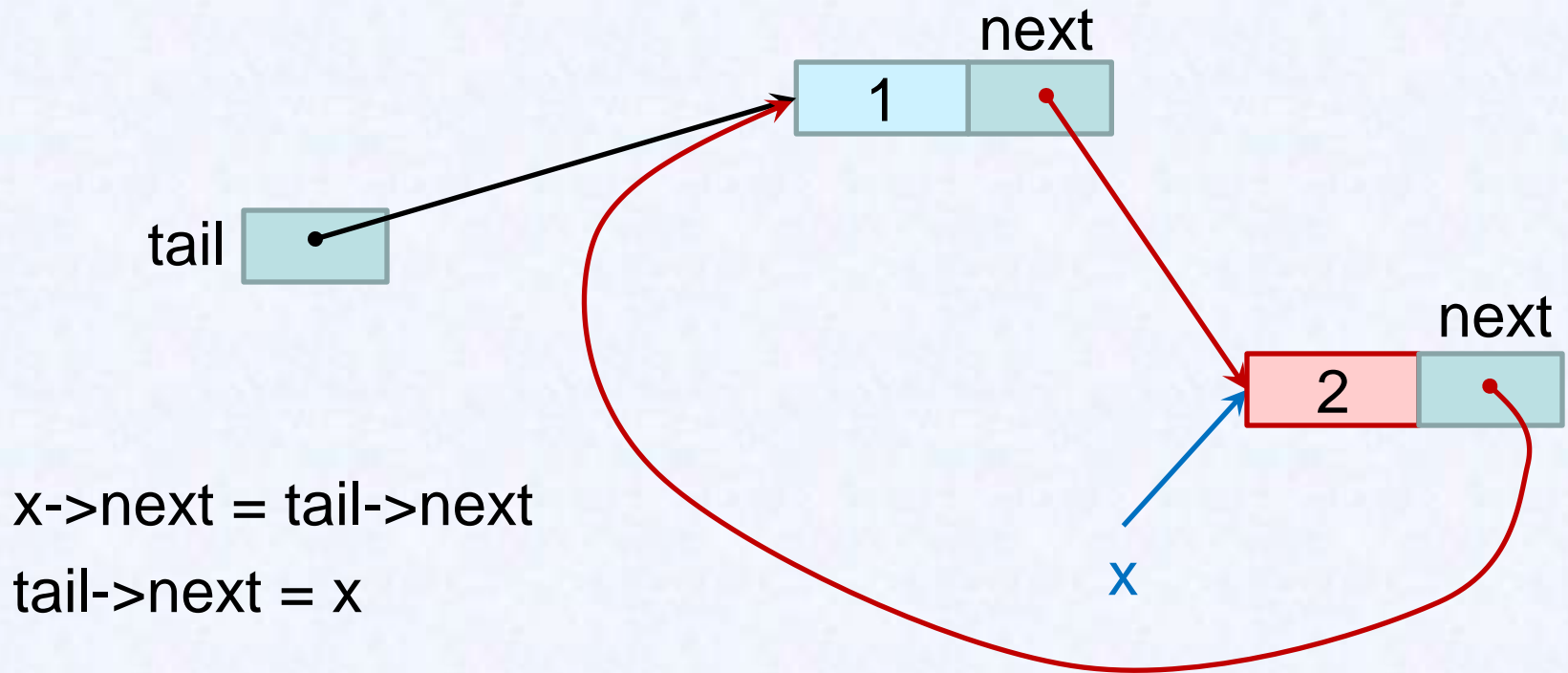
Запись в очередь – циклический список

Запись в очередь



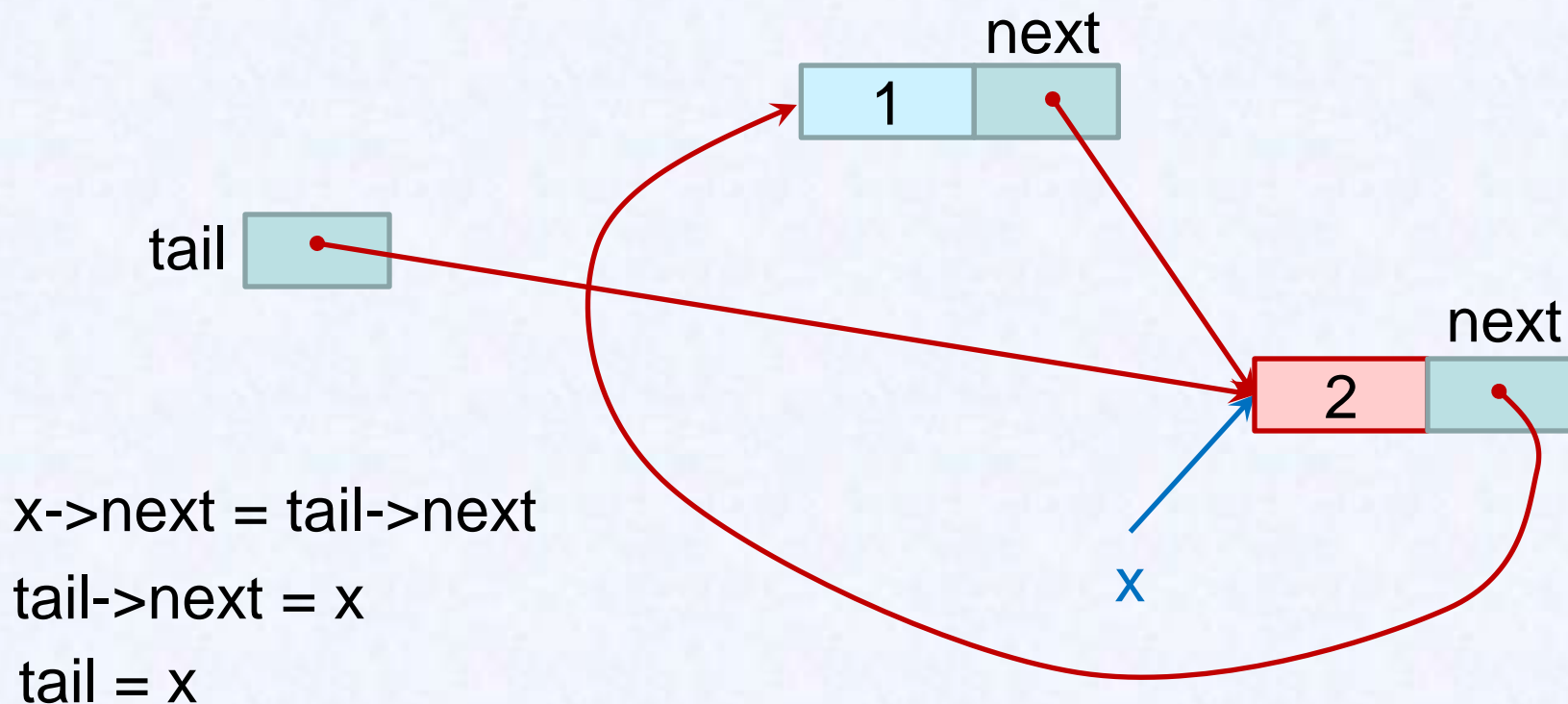
Запись в очередь – циклический список

Запись в очередь

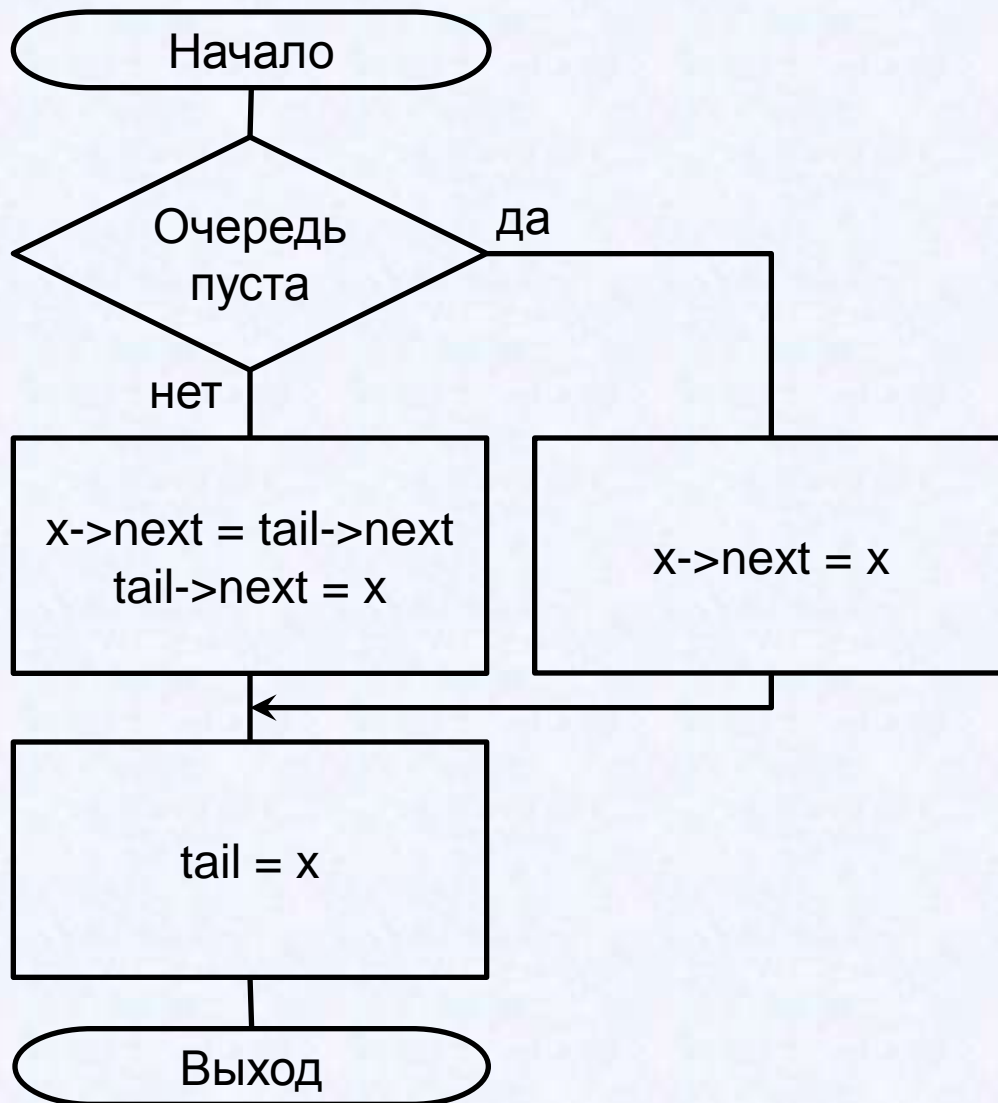


Запись в очередь – циклический список

Запись в очередь



Запись в очередь – циклический список



if очередь пуста

$x \rightarrow \text{next} = x$

else {

$x \rightarrow \text{next} = \text{tail} \rightarrow \text{next}$

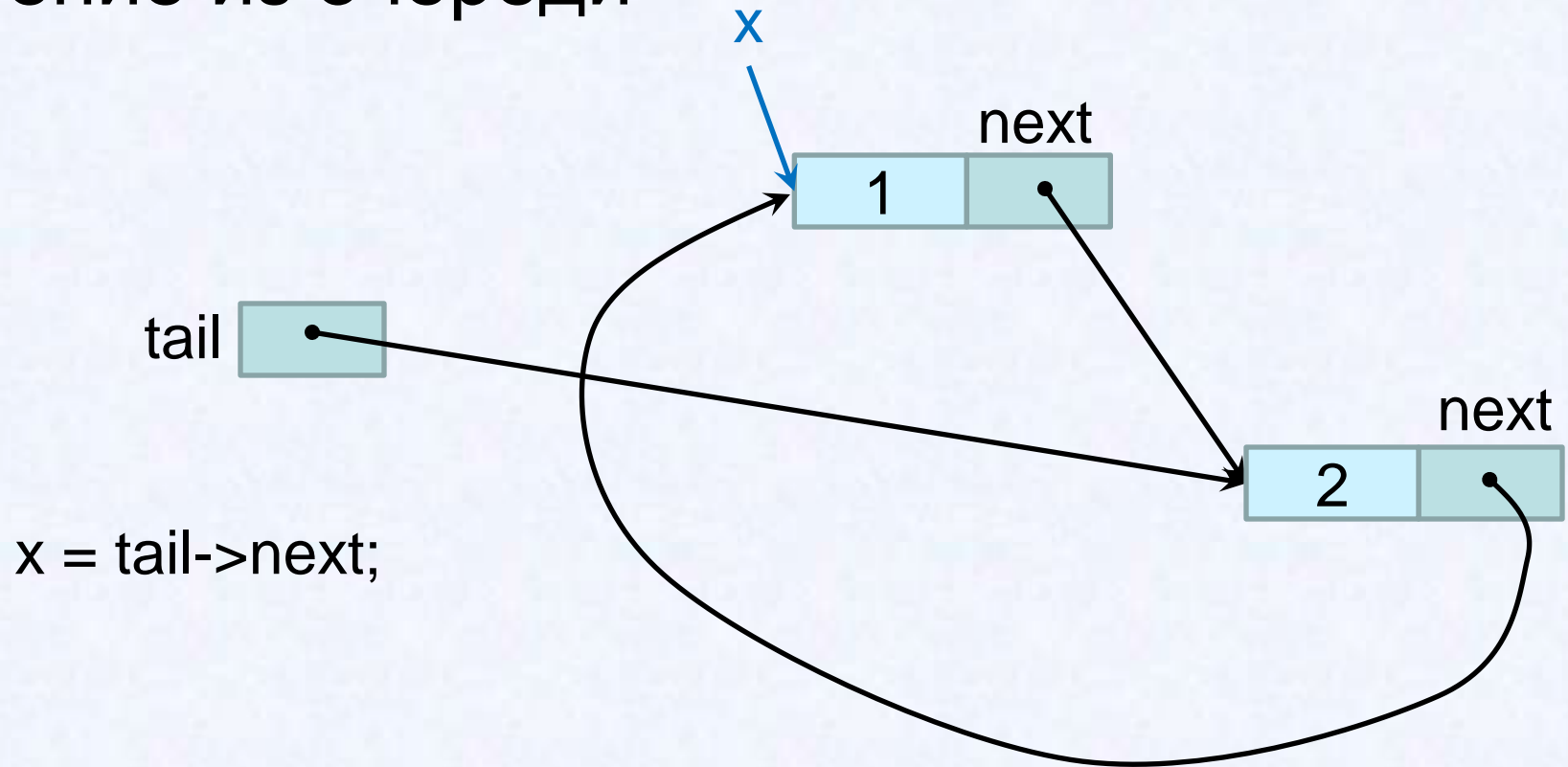
$\text{tail} \rightarrow \text{next} = x$

}

$\text{tail} = x$

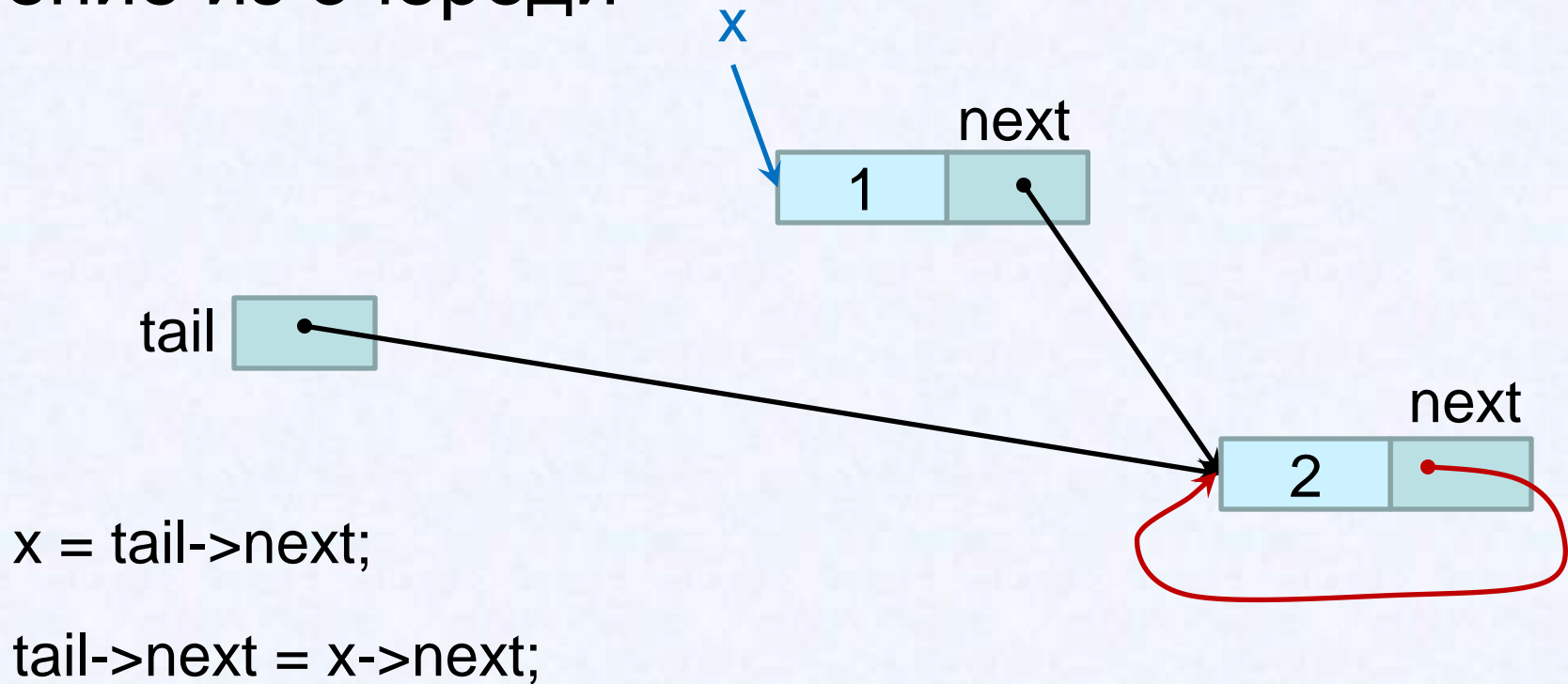
Чтение из очереди – циклического списка

Чтение из очереди



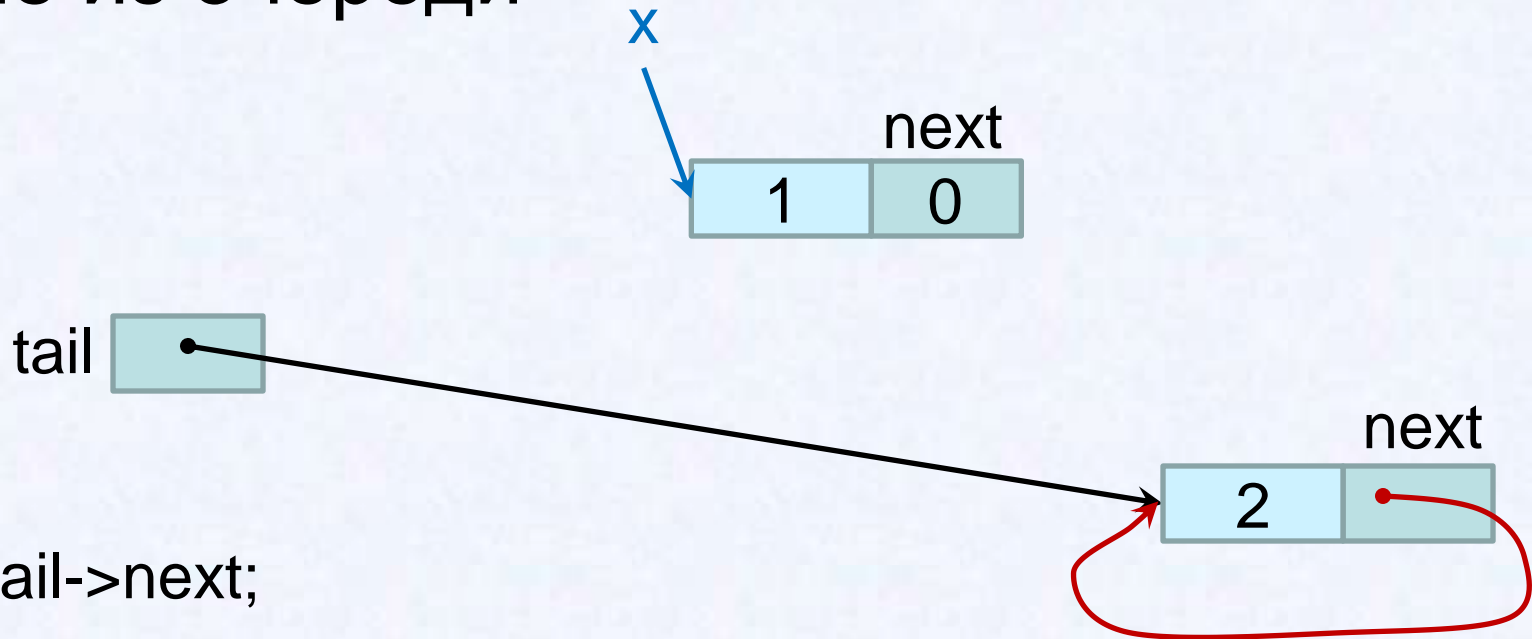
Чтение из очереди – циклического списка

Чтение из очереди



Чтение из очереди – циклического списка

Чтение из очереди



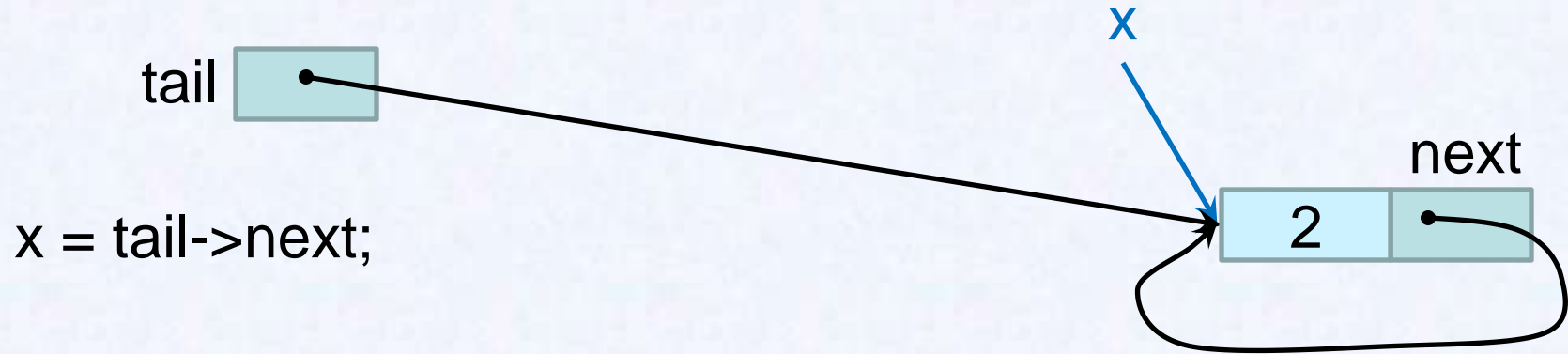
```
x = tail->next;
```

```
tail->next = x->next;
```

```
x->next = 0;
```


Чтение из очереди – циклического списка

Чтение из очереди



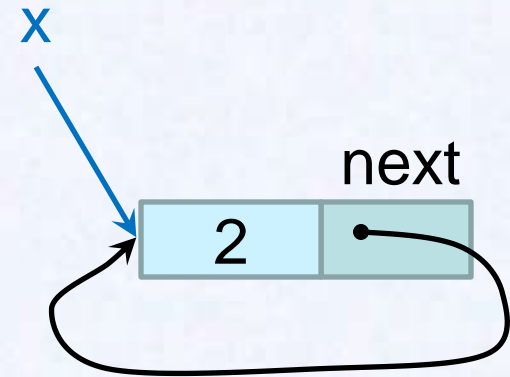
Чтение из очереди – циклического списка

Чтение из очереди

tail 0

`x = tail->next;`

tail->next совпадает с tail → **tail = 0**



Чтение из очереди – циклического списка

Чтение из очереди

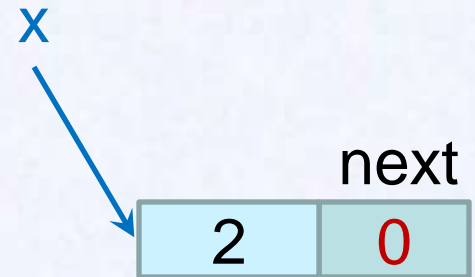
tail

0

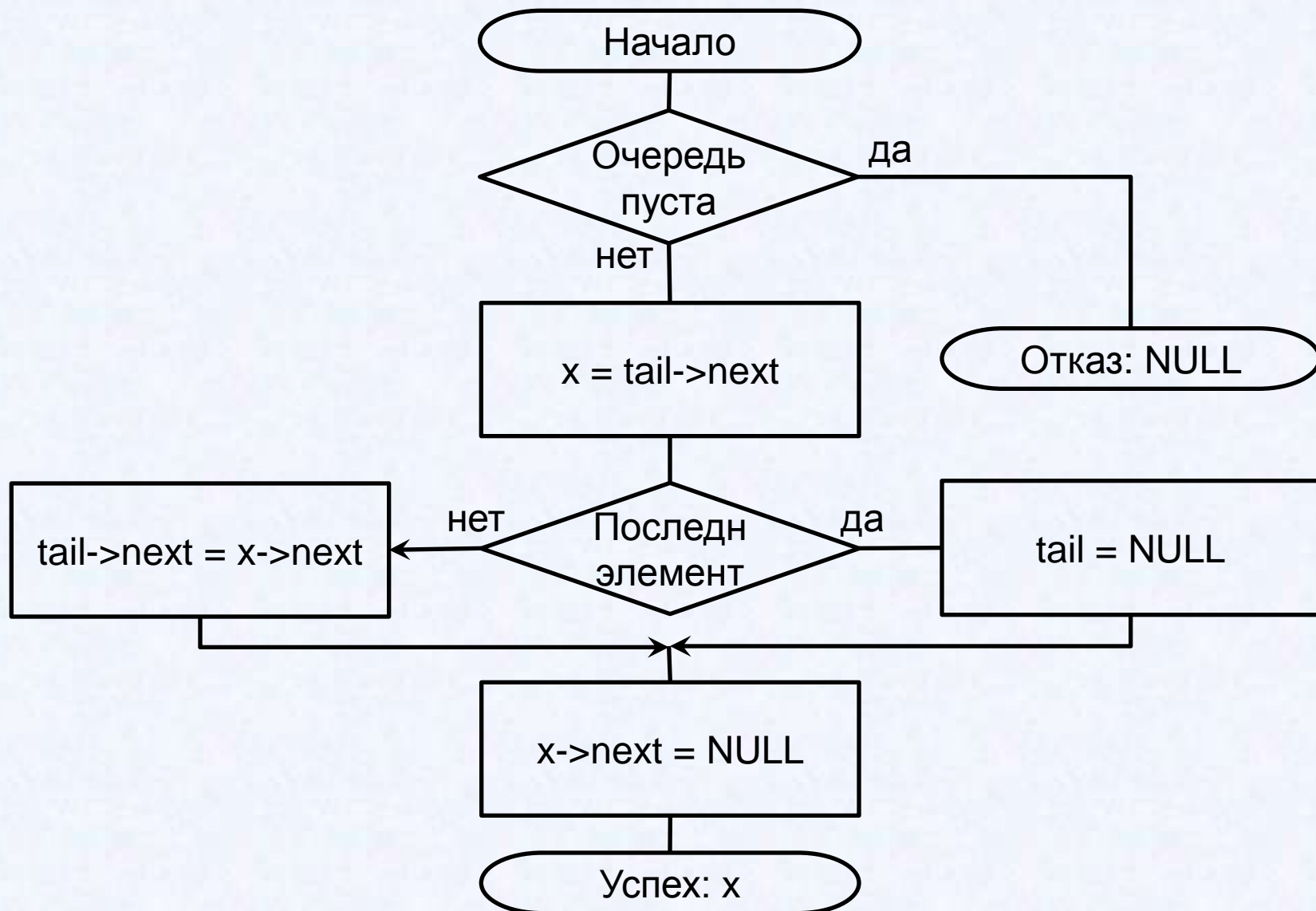
`x = tail->next;`

`tail->next` совпадает с `tail` → **tail = 0**

`x->next = 0;`



Чтение из очереди – циклического списка



Чтение из очереди – циклического списка

if очередь пуста

Отказ

`x = tail->next`

if удаляется последний элемент

`tail = NULL`

`else`

`tail->next = x->next`

`x->next = NULL`