Однопроходные алгоритмы

- Если последовательность можно не хранить – ее хранить не нужно!

Задача 1

- В последовательности записаны целые числа от 1 до N в произвольном порядке, но одно из чисел пропущено (остальные встречаются ровно по одному разу). N заранее неизвестно. Определить пропущенное число

O(N) – время
O(1) – память

Задача 1. Обсуждение

- В последовательности записаны целые числа от 1 до N в произвольном порядке, но одно из чисел пропущено (остальные встречаются ровно по одному разу). N заранее неизвестно. Определить пропущенное число.

sum = 19
sum = $\frac{1+6}{2} \cdot 6 = 14$ 2.

1. sort – $N \log N$
1 2 3 4

2

- Ввод: 1 6 5 4 3
- Вывод: 2

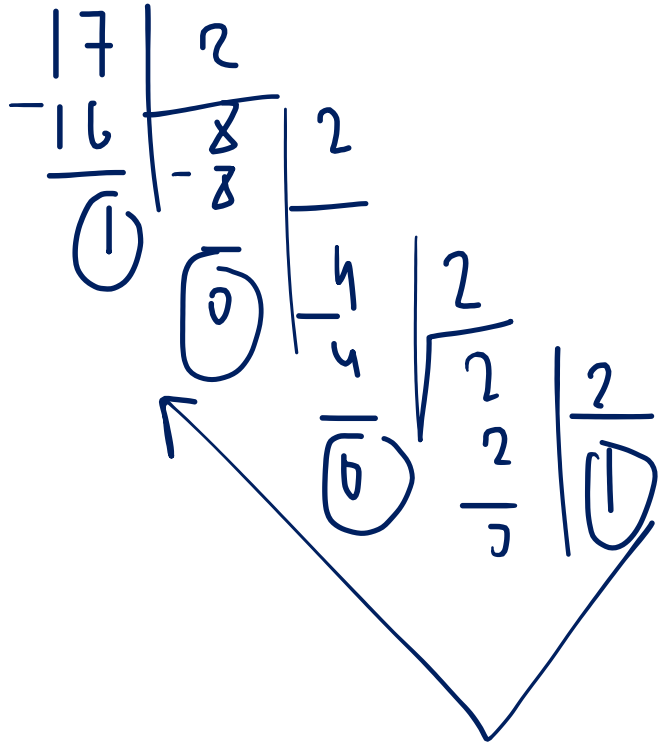
- Ввод: 1 4 3 2
- Вывод: 5

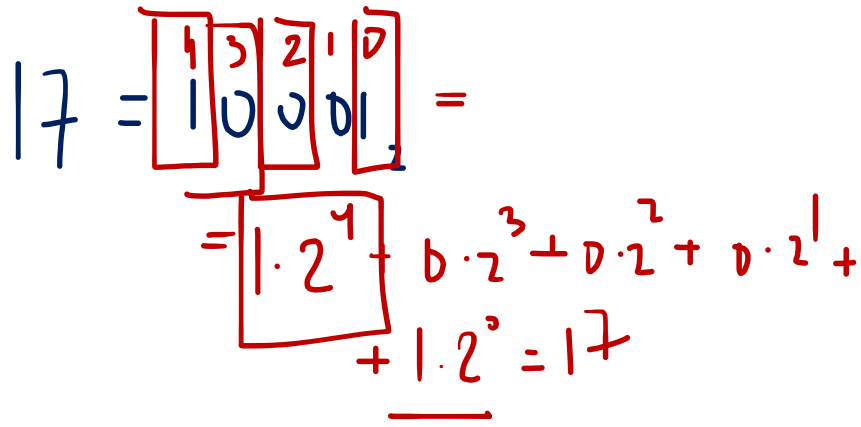
Задача 1. Решение

- В последовательности записаны целые числа от 1 до N в произвольном порядке, но одно из чисел пропущено (остальные встречаются ровно по одному разу). N заранее неизвестно. Определить пропущенное число.

- Ввод: 1 6 5 4 3
- Вывод: 2

- Ввод: 1 4 3 2
- Вывод: 5





Задача 2

- В последовательности записаны целые числа. Одно из чисел встречается ровно один раз, остальные — по два раза. Найти число, которое встречается один раз.

- Ввод: 1 2 2 1 3 4 3
- Вывод: 4

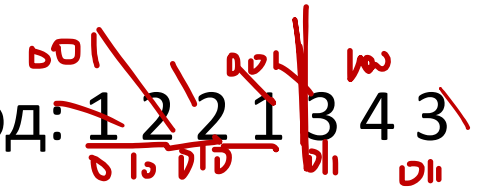
- Ввод: 1 1 2 9 10 9 2
- Вывод: 10

```
l = [1, 6, 5, 4, 3]
max_l = max(l)
n = len(l) + 1
if (max_l == len(l)):
  max_l += 1
sum_l = sum(l)
sum_l_2 = (1 + max_l) * n // 2
print(sum_l_2 - sum_l)
```

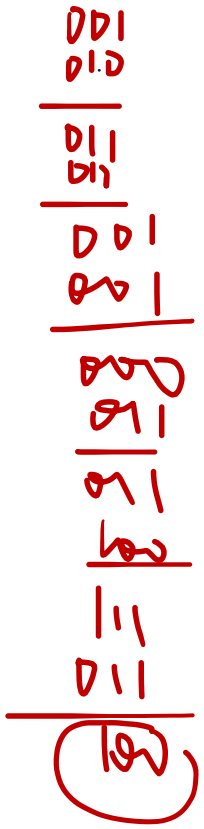
```
l = [1, 4, 3, 2]
max_l = 0
n = len(l) + 1
sum_l = 0
for i in range(len(l)):
  sum_l += l[i]
  max_l = max(max_l, l[i])
if (max_l == len(l)):
  max_l += 1
sum_l_2 = (1 + max_l) * n // 2
print(sum_l_2 - sum_l)
```

Задача 2. Обсуждение

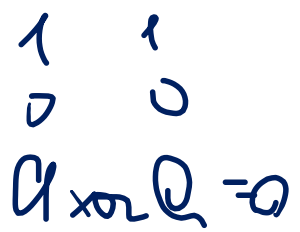
- В последовательности записаны целые числа. Одно из чисел встречается ровно один раз, остальные — по два раза. Найти число, которое встречается один раз.

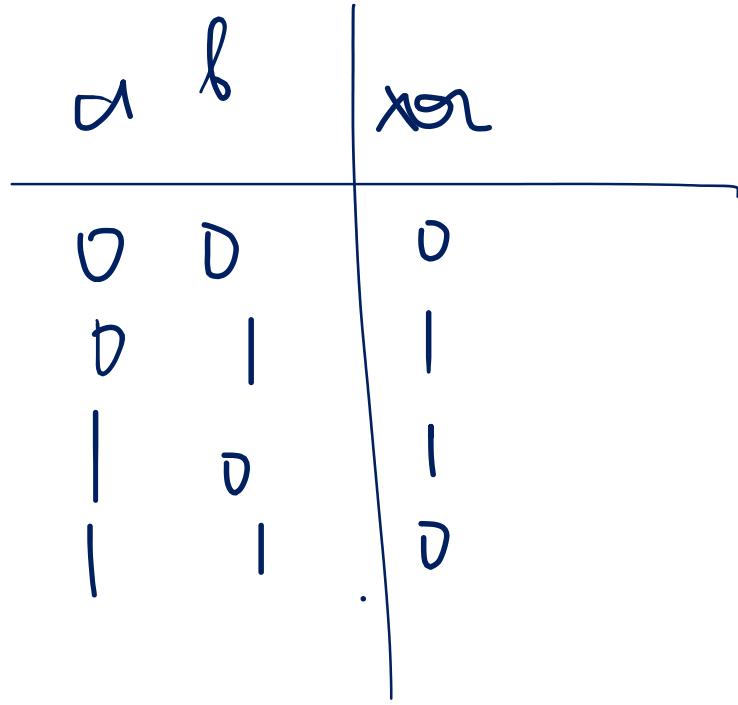


- Ввод: 1 2 2 1 3 4 3
- Вывод: 4



- Ввод: 1 1 2 9 10 9 2
- Вывод: 10





Задача 2. Решение

- В последовательности записаны целые числа. Одно из чисел встречается ровно один раз, остальные — по два раза. Найти число, которое встречается один раз.

- Ввод: 1 2 2 1 3 4 3
- Вывод: 4

- Ввод: 1 1 2 9 10 9 2
- Вывод: 10

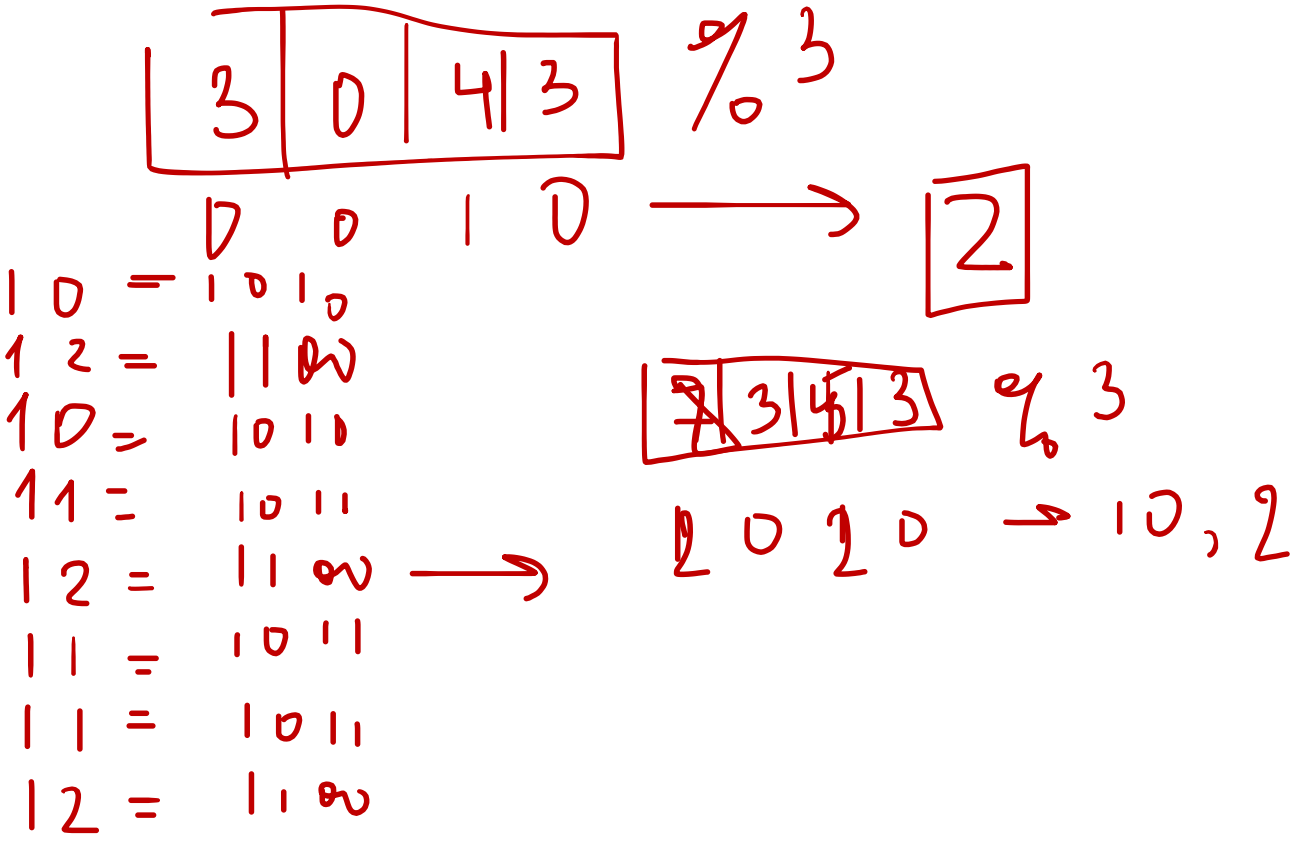
```
l = [1, 2, 2, 1, 3, 4, 3]
num = 0
for i in range(len(l)):
  num ^= l[i]
print(num)
```

Задача 3

- В последовательности записаны целые числа. Число X встречается один или два раза, остальные числа — по три раза. Найти число X и количество его встреч. Для простоты считаем, что числа неотрицательные.

- Ввод: 1 2 2 1 2 1 3
- Вывод: 3, 2

- Ввод: 1 2 10 10 10 1 1
- Вывод: 2, 1



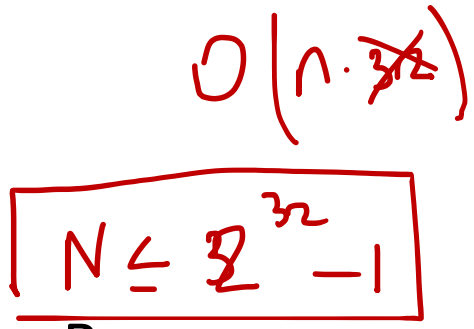
Задача 3. Обсуждение

- В последовательности записаны целые числа. Число X встречается один или два раза, остальные числа — по три раза. Найти число X и количество его встреч. Для простоты считаем, что числа неотрицательные.

- Ввод: 1 2 2 1 2 1 3
- Вывод: 3, 2

- Ввод: 1 2 10 10 10 1 1
- Вывод: 2, 1

Задача 3. Решение



- В последовательности записаны целые числа. Число X встречается один или два раза, остальные числа — по три раза. Найти число X и количество его встреч. Для простоты считаем, что числа неотрицательные.

- Ввод: 1 2 2 1 2 1 3
- Вывод: 3, 2

- Ввод: 1 2 10 10 10 1 1
- Вывод: 2, 1

```
l = [1, 2, 2, 1, 2, 1, 3, 3]
bin = [0] * 32
for i in range(len(l)):
    num = l[i]
    j = 0
    while (num != 0):
        bin[j] += num % 2
        num //= 2
        j += 1
```

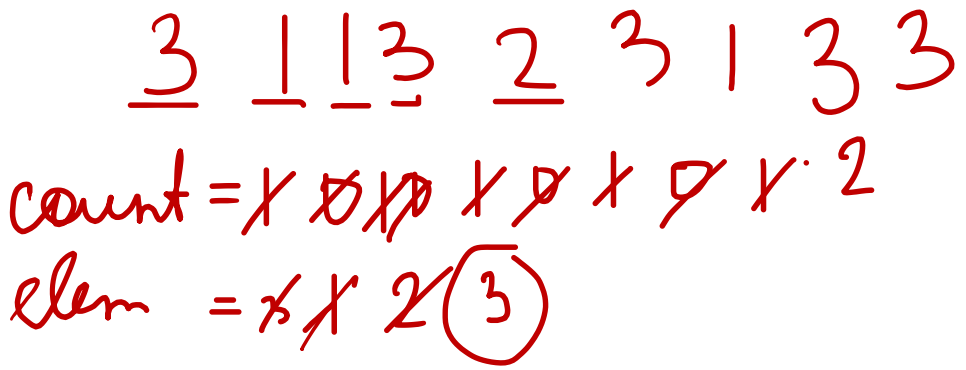
```
num = 0
pow_2 = 1
count = 0
for i in range(len(bin)):
    bin[i] %= 3
    if (bin[i] > 0):
        num += pow_2
        count = bin[i]
    pow_2 *= 2
print(num, count)
```

Задача 4.

- В последовательности записаны целые числа, больше половины из которых равны одному и тому же числу X. За один просмотр последовательности найти это число.

- Ввод: 1 2 2 1 2 1 1 1 1 1 1
- Вывод: 1

- Ввод: 1 2 100 100 100 100
- Вывод: 100



Задача 4. Обсуждение

- В последовательности записаны целые числа, больше половины из которых равны одному и тому же числу X. За один просмотр последовательности найти это число.

- Ввод: 1 2 2 1 2 1 1 1 1 1 1
- Вывод: 1

- Ввод: 1 2 100 100 100 100
- Вывод: 100

Задача 5. Решение

- В последовательности записаны целые числа, больше половины из которых равны одному и тому же числу X. За один просмотр последовательности найти это число.

- Ввод: 1 2 2 1 2 1 1 1 1 1 1
- Вывод: 1

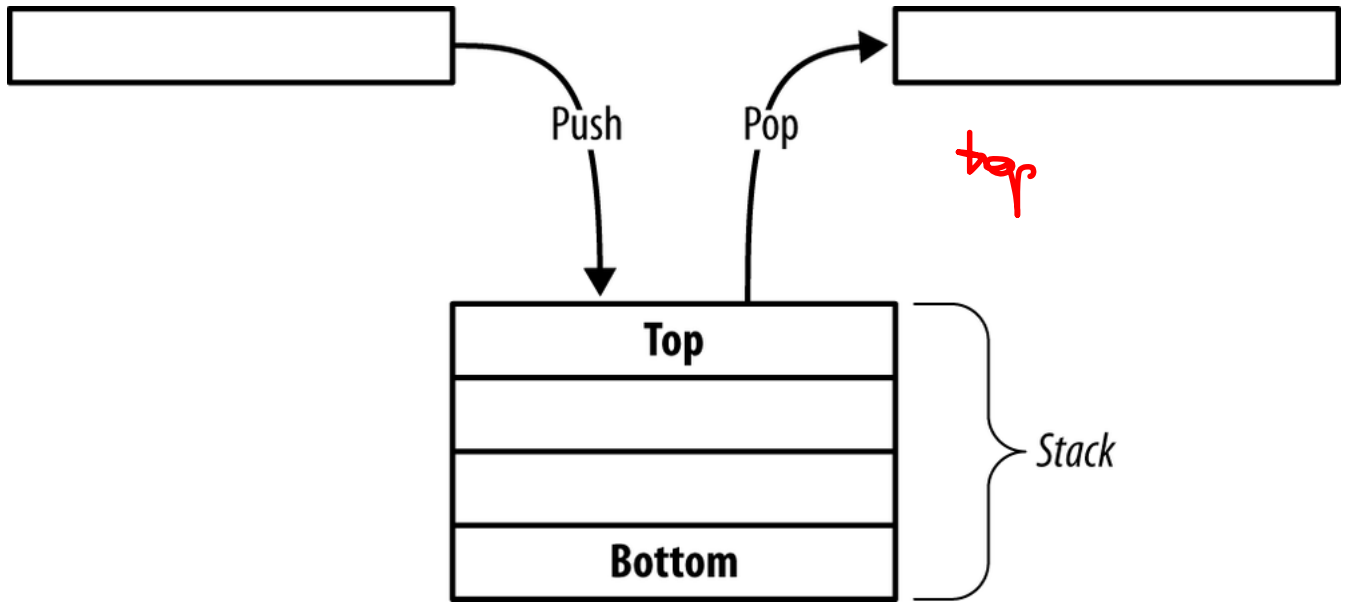
- Ввод: 1 2 100 100 100 100
- Вывод: 100

```
l = [1, 2, 100, 100, 100]
elem = 0
count = 0

for i in range(len(l)):
    if (count == 0):
        elem = l[i]
        count += 1
    elif (elem == l[i]):
        count += 1
    else:
        count -= 1
print(elem)
```

Stack

- **Стек** — это структура данных, которая работает по принципу **FILO** (first in — last out; первый пришел — последний ушел) или *(Last-In-First-Out или **LIFO**)*.

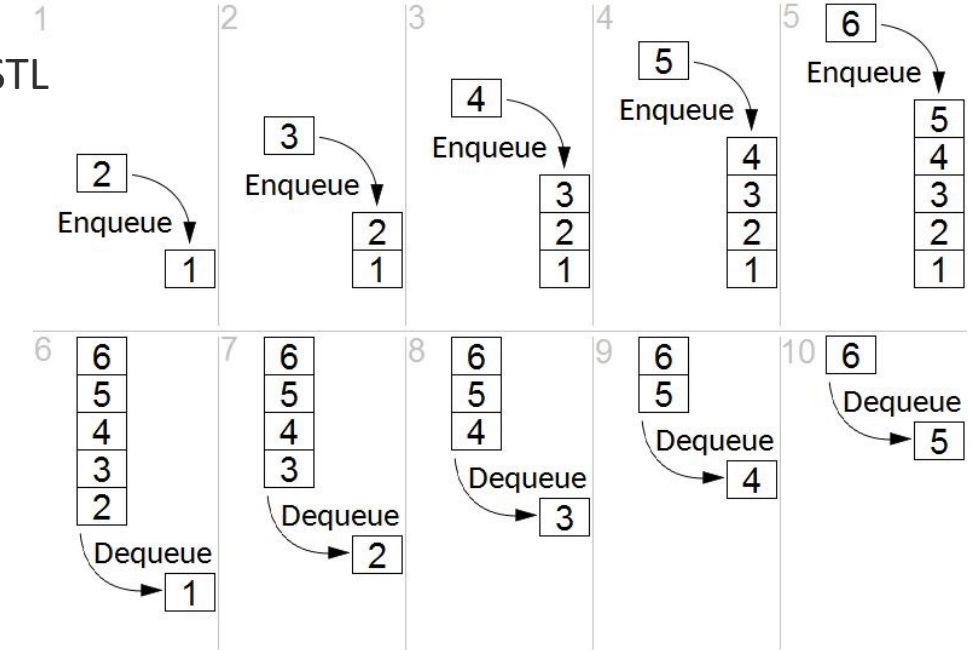


http://www.8bytes.net/2017/10/24/struktura-dannyh-ochered-fifo/

Queue

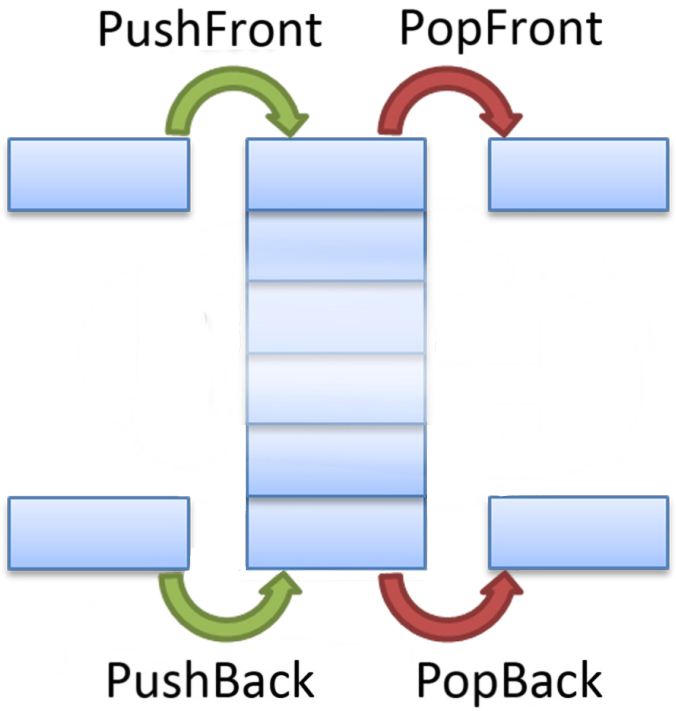
- **Очередью** (англ. – queue) называется структура данных, из которой удаляется первым тот элемент, который был первым в очередь добавлен. То есть очередь в программировании соответствует «бытовому» понятию очереди. Очередь также называют структурой типа FIFO (first in, first out — первым пришел, первым ушел).

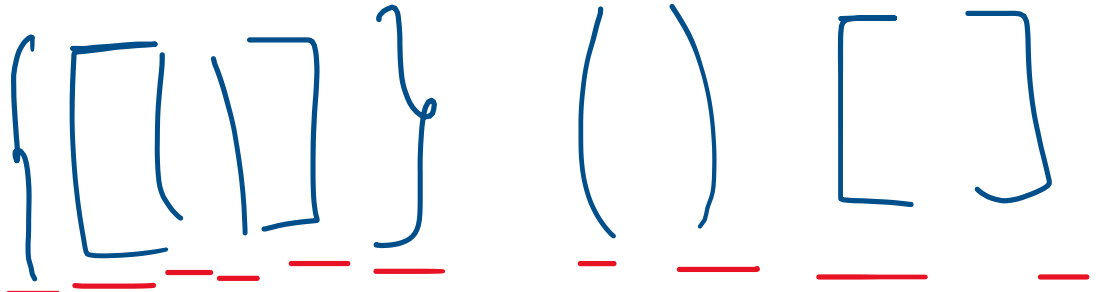
В C++ уже есть готовый STL контейнер — queue.

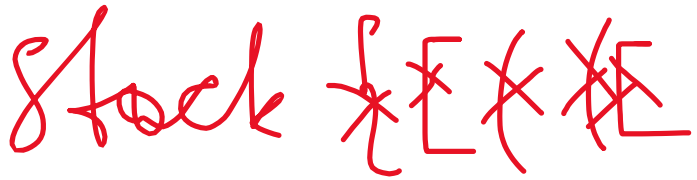


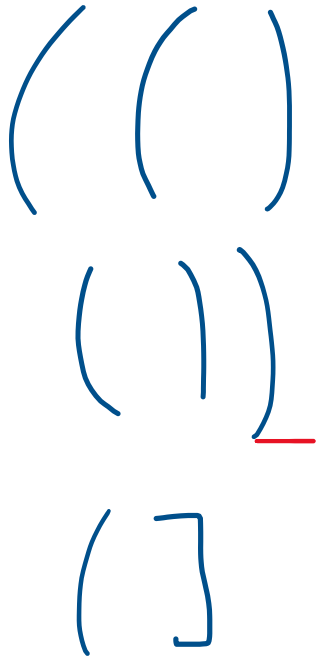
Deque

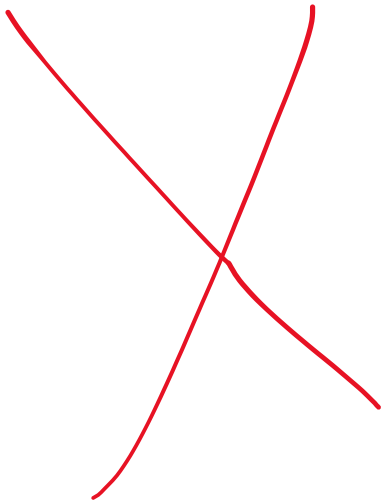
- **Двусвязная очередь** — абстрактный тип данных, в котором элементы можно добавлять и удалять как в начало, так и в конец.











- B C + D *

(B + C) ↘ b

- 8 9 + 1 7 - *

Stack: 8 9 17 x 7

-6 -102

- B + (C * D)
- A B C + D * +

A + (B+C) ↘ D

- 4
- 10 11 12 13