

# Алгоритмы поиска, очереди, стек

Семинар



# Является ли одна строка подпоследовательностью другой строки

В исходную строку добавили некоторое количество символов. Необходимо выявить, является ли строка  $a$  подстрокой  $b$ . Или другими словами была ли она исходной для  $b$ .



Является ли одна строка  
подпоследовательностью  
другой строки?

Используя очередь  $O(2n)$

```
function isSubsequence(a, b) {  
    q = Queue()  
    for el in a {  
        q.push(el)  
    }  
}
```

Является ли одна строка  
подпоследовательностью  
другой строки?

Используя очередь  $O(2n)$

```
function isSubsequence(a, b) {  
    q = Queue()  
    for el in a {  
        q.push(el)  
    }  
    for el in b {  
        ...  
    }  
}
```

Является ли одна строка  
подпоследовательностью  
другой строки?

Используя очередь  $O(2n)$

```
function isSubsequence(a, b) {  
    q = Queue()  
    for el in a {  
        q.push(el)  
    }  
    for el in b {  
        if q.peek() == el {  
            q.pop()  
        }  
    }  
}
```

Является ли одна строка  
подпоследовательностью  
другой строки?

Используя очередь  $O(2n)$

```
function isSubsequence(a, b) {  
    q = Queue()  
    for el in a {  
        q.push(el)  
    }  
    for el in b {  
        if q.peek() == el {  
            q.pop()  
        }  
    }  
    return q.getSize() == 0  
}
```

Является ли одна строка  
подпоследовательностью  
другой строки?

Методом двух указателей  $O(n)$

```
function isSubsequence(a, b) {  
    i, j = 0, 0  
}
```

Является ли одна строка  
подпоследовательностью  
другой строки?

Методом двух указателей  $O(n)$

```
function isSubsequence(a, b) {  
    i, j = 0, 0  
    while i < len(a) and j < len(b) {  
        if a[i] == b[j] {  
            i++  
        }  
        j++  
    }  
}
```



Является ли одна строка  
подпоследовательностью  
другой строки?

Методом двух указателей  $O(n)$

```
function isSubsequence(a, b) {  
    i, j = 0, 0  
    while i < len(a) and j < len(b) {  
        if a[i] == b[j] {  
            i++  
        }  
        j++  
    }  
    return i == len(a)  
}
```

Является ли одна строка  
подпоследовательностью  
другой строки?

Наивная реализация  $O(n^2)$

```
i = 0
for el1 in a {
    for el2 in b {
        if el1 == el2 {
            i++
            break
        }
    }
}
```

Является ли слово  
палиндромом



# Является ли слово полиндромом?

Используя стек

```
function isPalindrome(s) {  
    stack = list()  
    for char in s {  
        stack.push(char)  
    }  
}
```

# Является ли слово полиндромом?

Используя стек

```
function isPalindrome(s) {  
    stack = list()  
    for char in s {  
        stack.push(char)  
    }  
    for char in s {  
        if char != stack.pop() {  
            return false  
        }  
    }  
}
```

# Является ли слово полиндромом?

Используя стек

```
function isPalindrome(s) {  
    stack = list()  
    for char in s {  
        stack.push(char)  
    }  
    for char in s {  
        if char != stack.pop() {  
            return false  
        }  
    }  
    return true  
}
```

# Является ли слово полиндромом?

Методом двух указателей

```
function isPalindrome(s) {  
    l = 0  
    r = len(s) - 1  
}
```

# Является ли слово полиндромом?

Методом двух указателей

```
function isPalindrome(s) {  
    l = 0  
    r = len(s) - 1  
    while l <= r {  
        l++  
        r--  
    }  
}
```



# Является ли слово палиндромом?

Методом двух указателей

```
function isPalindrome(s) {  
    l = 0  
    r = len(s)-1  
    while l <= r {  
        if s[l] != s[r] {  
            return false  
        }  
        l++  
        r--  
    }  
    return true  
}
```

Найти корень числа  
(ближайшее целое)



Найти корень числа  
(ближайшее целое)

```
function binarySearchSqrt(target) {  
    l = 0; r = target  
}
```

Найти корень числа  
(ближайшее целое)

```
function binarySearchSqrt(target) {  
    l = 0; r = target  
    ...  
    return r  
}
```

# Найти корень числа (ближайшее целое)

```
function binarySearchSqrt(target) {  
    l = 0; r = target  
    while l <= r {  
        middle = (l+r) / 2  
        if middle2 > target {  
            r = middle-1  
            continue  
        }  
    }  
}
```

# Найти корень числа (ближайшее целое)

```
function binarySearchSqrt(target) {  
    l = 0; r = target  
    while l <= r {  
        middle = (l+r) / 2  
        if middle2 > target {  
            r = middle-1  
            continue  
        }  
        if middle2 < target {  
            l = middle+1  
            continue  
        }  
        return middle  
    }  
}
```

# Найти корень числа (ближайшее целое)

```
function binarySearchSqrt(target) {  
    l = 0; r = target  
    while l <= r {  
        middle = (l+r) / 2  
        if middle2 > target {  
            r = middle-1  
            continue  
        }  
        if middle2 < target {  
            l = middle+1  
            continue  
        }  
        return middle  
    }  
    return r  
}
```

# Очень лёгкая задача

Как быстро можно сделать  $N$  копий документа, используя два ксерокса, каждый со своей скоростью ( $x$  и  $y$  минут)?





# Очень лёгкая задача

Сегодня утром жюри решило добавить в вариант олимпиады еще одну, Очень Легкую Задачу. Ответственный секретарь Оргкомитета напечатал ее условие в одном экземпляре, и теперь ему нужно до начала олимпиады успеть сделать еще  $N$  копий. В его распоряжении имеются два ксерокса, один из которых копирует лист за  $x$  секунд, а другой – за  $y$ . (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ему выяснить, какое минимальное время для этого потребуется.

```
function copyTime(n, x, y) {  
    l = 0  
    r = (n - 1) * max(x, y)  
}
```

# Очень лёгкая задача

Сегодня утром жюри решило добавить в вариант олимпиады еще одну, Очень Легкую Задачу. Ответственный секретарь Оргкомитета напечатал ее условие в одном экземпляре, и теперь ему нужно до начала олимпиады успеть сделать еще  $N$  копий. В его распоряжении имеются два ксерокса, один из которых копирует лист за  $x$  секунд, а другой – за  $y$ . (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ему выяснить, какое минимальное время для этого потребуется.

```
function copyTime(n, x, y) {  
    l = 0  
    r = (n - 1) * max(x, y)  
  
    while l + 1 < r {  
        mid = (r + l) / 2  
    }  
}
```

# Очень лёгкая задача

Сегодня утром жюри решило добавить в вариант олимпиады еще одну, Очень Легкую Задачу. Ответственный секретарь Оргкомитета напечатал ее условие в одном экземпляре, и теперь ему нужно до начала олимпиады успеть сделать еще  $N$  копий. В его распоряжении имеются два ксерокса, один из которых копирует лист за  $x$  секунд, а другой – за  $y$ . (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ему выяснить, какое минимальное время для этого потребуется.

```
function copyTime(n, x, y) {  
    l = 0  
    r = (n - 1) * max(x, y)  
  
    while l + 1 < r {  
        mid = (r + l) / 2  
        if mid/x + mid/y < n-1 {  
            l = mid  
        } else {  
            r = mid  
        }  
    }  
}
```

# Очень лёгкая задача

Сегодня утром жюри решило добавить в вариант олимпиады еще одну, Очень Легкую Задачу. Ответственный секретарь Оргкомитета напечатал ее условие в одном экземпляре, и теперь ему нужно до начала олимпиады успеть сделать еще  $N$  копий. В его распоряжении имеются два ксерокса, один из которых копирует лист за  $x$  секунд, а другой – за  $y$ . (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ему выяснить, какое минимальное время для этого потребуется.

```
function copyTime(n, x, y) {  
    l = 0  
    r = (n - 1) * max(x, y)  
  
    while l + 1 < r {  
        mid = (r + l) / 2  
        if mid/x + mid/y < n-1 {  
            l = mid  
        } else {  
            r = mid  
        }  
    }  
  
    return r + min(x, y)  
}
```

Всем  
спасибо!