

# Яндекс.Практикум, Алгоритмы: Спринт 1.

## Финальные задачи

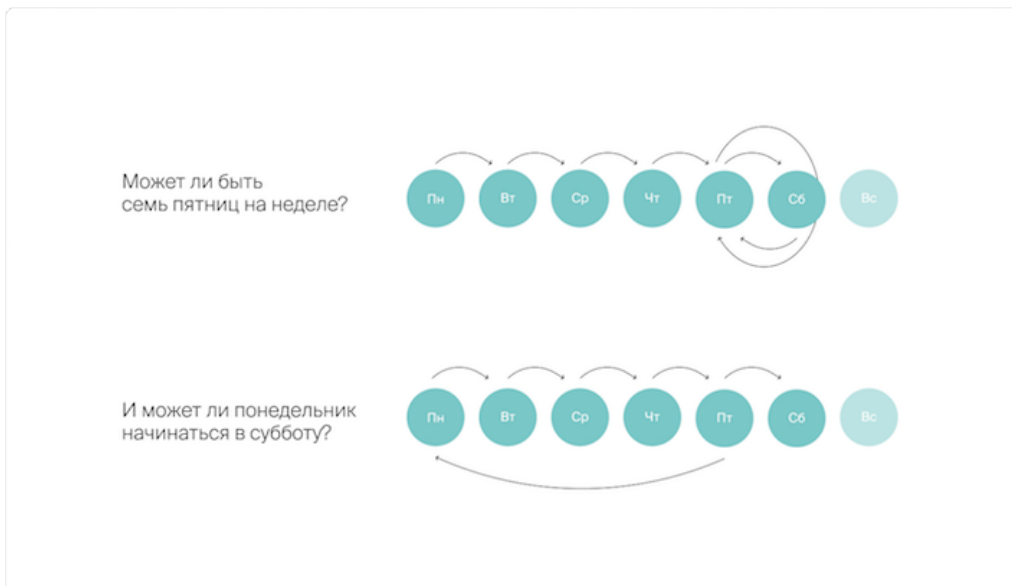
🕒 7 авг 2020, 00:54:19  
старт: 5 авг 2020, 01:56:18

### В. Циклы

Ограничение времени	0.65 секунд
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

В этом спринте вы изучили структуру данных Связный список.

В Связном списке можно, например, хранить дни недели. Вторник за понедельником, среда за вторником, и так далее.



Вам предстоит это выяснить!

В качестве второго задания финального проекта нужно написать программу, которая определяет, есть ли цикл в связном списке.

На вход функция принимает голову списка, на выходе должна выдать True, если в списке содержится цикл, иначе — False. Размер дополнительной памяти, к которой обращается функция, не должен превышать  $O(1)$ .

### Формат ввода

В этой задаче вам нужно реализовать только функцию с решением, считывать входные данные не нужно.

Функция должна принимать на вход голову связного списка.

Класс / структура, представляющая узел списка выглядит так:

```
C++:  
struct Node  
{  
    std::weak_ptr<Node> next;  
};
```

Ваша функция должна называться HasCycle.

Сигнатура функции: `bool HasCycle(std::shared_ptr<Node> head)`

Нужно подключить solution.h

Go:

```
type TListItem struct {  
    value    string
```

```
nextElement *TListItem
}
```

Ваша функция должна называться HasCycle. Нужно подключить package main

JS:

```
class Node {
  constructor(value) {
    this.value = value;
    this.next = null;
  }
}
```

Ваша функция должна называться hasCycle.

Python:

```
class Node:
    def __init__(self, value, next=None):
        self.value = value
        self.next = next
    def __repr__(self):
        return self.value
```

Ваша функция должна называться hasCycle.

Java:

Файл должен содержать public class Solution с public static String hasCycle (Node head)

```
class Node<V> {
    private V value;
    private Node<V> next;
    public Node(V value, Node<V> next) {
        this.value = value;
        this.next = next; }
    public V getValue() {
        return value; }
    public Node<V> getNext() {
        return next; }
    public void setNext(Node<V> next) {
        this.next = next; }
    public boolean hasNext() {
        return next != null; }
    public void setValue(V value) {
        this.value = value; }
}
```

## Формат вывода

Для всех языков, кроме JS и Go функция должна возвращать булево значение: True/true или False/false

Для JS и Go - строку 'True' или 'False'

## Примечания

Можно использовать только  $O(1)$  дополнительной памяти.

При отправке нужно выбирать компилятор make и загружать решение в виде файла. Для всех языков программирования, кроме Java, файл может быть назван любым именем, кроме solution.py/cpp/go/js (расширение обязательно)

Для Java файл должен называться Solution.java

Язык Make ▼

Набрать здесь

Отправить файл

Выбрать

Файл не выбран

Отправить

Предыдущая

Следующая

