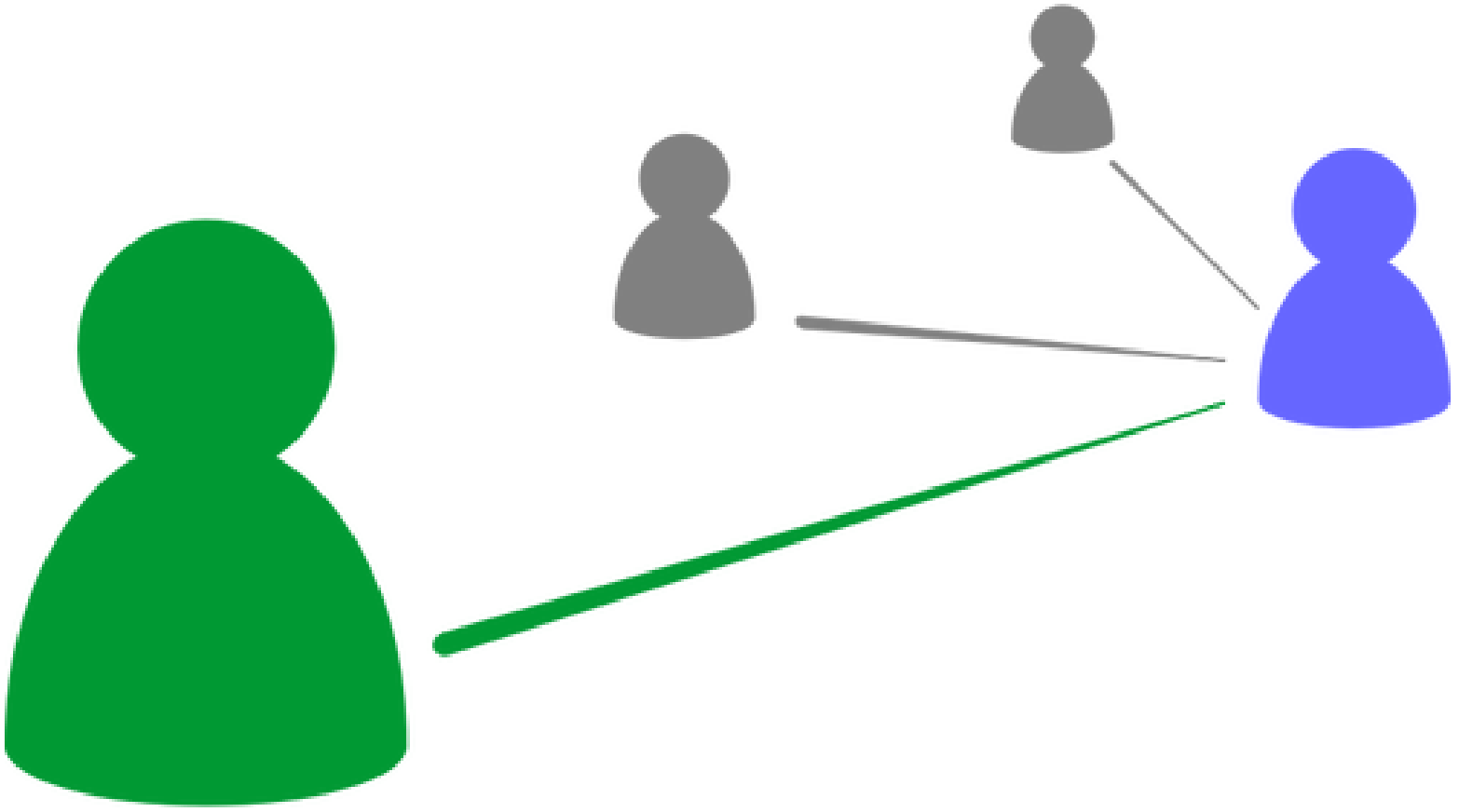
ОГЛАВЛЕНИЕ

Знакомство

Каждый представляется и рассказывает из какой области пришел.



Что такое Алгоритм

Алгоритм означает набор правил, которым необходимо следовать при вычислениях или других операциях по решению задач.

Алгоритм относится к последовательности конечных шагов для решения конкретной проблемы.

Алгоритмы могут быть простыми и сложными в зависимости от того, чего вы хотите достичь.

Алгоритмизация – процесс разработки алгоритма для решения какой-либо задачи.

Характеристики алгоритма

Ясный и недвусмысленный: каждый его шаг должен быть ясен во всех аспектах и должен вести только к одному смыслу.

Четко определенные входные данные: если алгоритм говорит принимать входные данные, это должны быть четко определенные входные данные.

Четко определенные результаты: Алгоритм должен четко определять, какой результат будет получен, и он также должен быть четко определен.

Конечность: Алгоритм должен быть конечным, т.е. он должен завершаться через конечное время.

Выполнимый: алгоритм должен быть простым, универсальным и практичным, чтобы его можно было выполнить с доступными ресурсами.

Независимый от языка: разработанный алгоритм должен быть независимым от языка, т. е. это должны быть простые инструкции, которые могут быть реализованы на любом языке, и при этом вывод будет таким же, как и ожидалось.

Свойства алгоритма

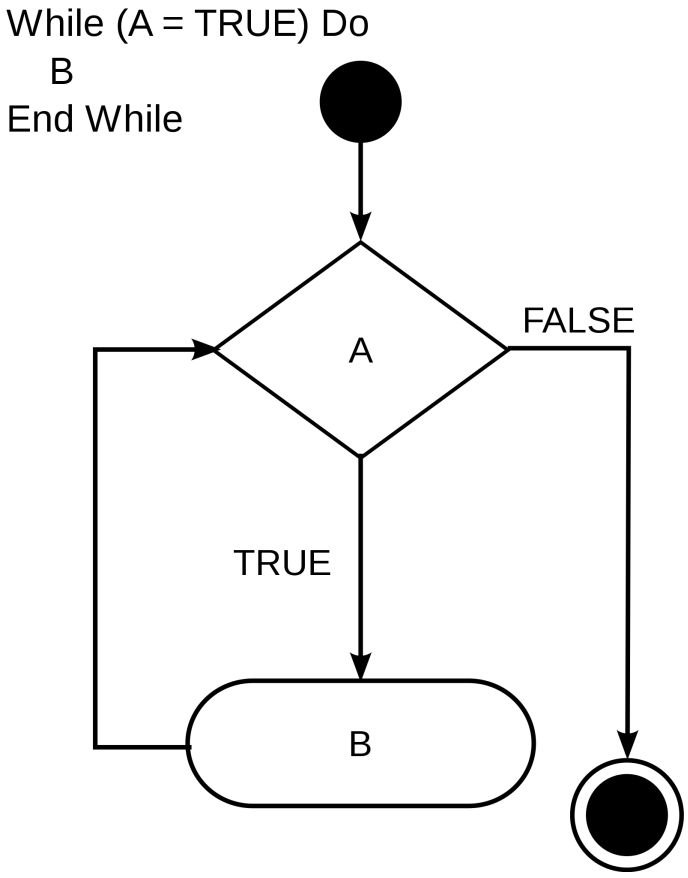
Алгоритм **должен** завершиться через конечное время.

Алгоритм **должен** производить хотя бы один вывод.

Алгоритм **должен** принимать ноль или более входных данных.

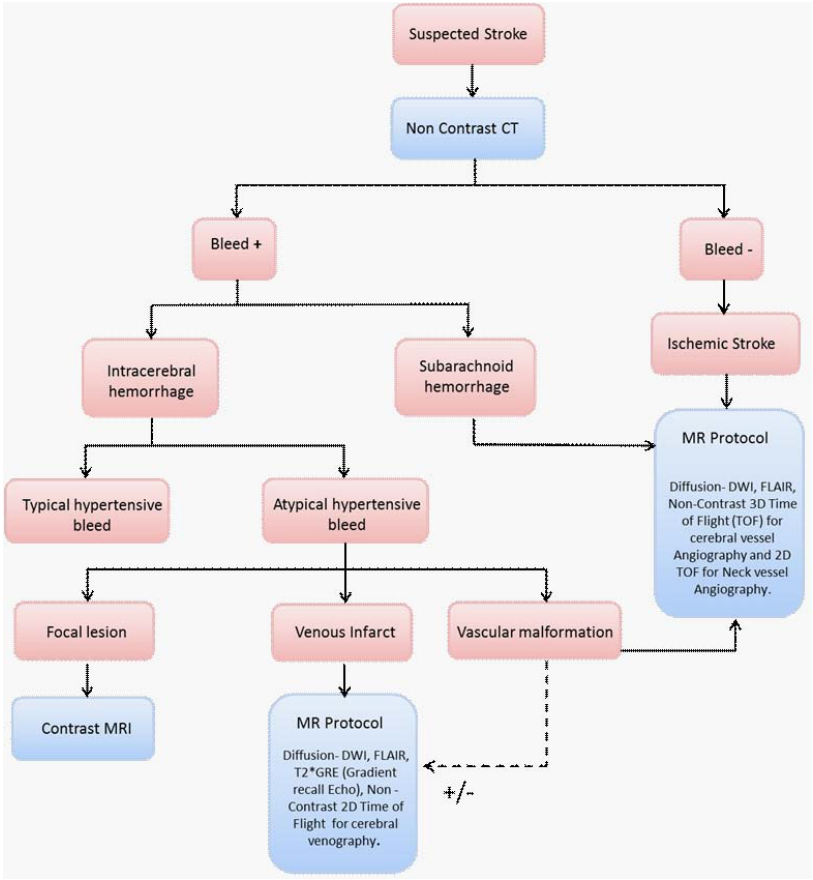
Алгоритм **должен** давать один и тот же результат для одного и того же входного случая.

Каждый шаг в алгоритме **должен** быть эффективным.



Типы алгоритмов

1. Алгоритм грубой силы
2. Рекурсивный алгоритм
3. Алгоритм поиска с возвратом
4. Алгоритм поиска
5. Алгоритм сортировки
6. Алгоритм хеширования
7. Алгоритм «разделяй и властвуй»
8. Жадный алгоритм
9. Алгоритм динамического программирования
10. Рандомизированный алгоритм

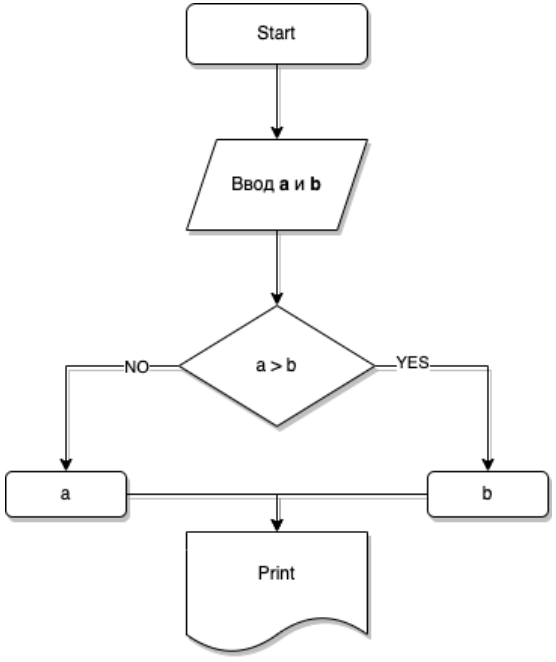


Как разработать алгоритм

1. Проблема, которая должна быть решена с помощью этого алгоритма, т.е. четкое определение проблемы.
2. При решении проблемы необходимо учитывать все ограничения.
3. Входные данные, которые необходимо принять для решения этой проблемы.
4. Ожидаемый результат после решения проблемы.
5. Решение этой проблемы находится в рамках заданных ограничений.

Существует три основных способа описания алгоритма:

- Текстовой – расписываете шаги алгоритма последовательно в тексте
- Алгоритмический язык – псевдокод
- Графический способ – изображается графически в виде блок-схем.



Блок схема

```
START
Input number: a, b
IF a > b then conclusion a
ELSE output b
END
```

Псевдокод

Эффективность алгоритма

Чтобы алгоритм был хорошим, он должен быть эффективным. Следовательно, эффективность алгоритма должна проверяться и поддерживаться.

Фактор времени: время измеряется путем подсчета количества ключевых операций. `Time Complexity`

Фактор пространства: пространство измеряется путем подсчета максимального объема памяти, требуемого алгоритмом. `Space Complexity`

Преимущества и недостатки алгоритмов

Преимущества алгоритмов:	**Недостатки алгоритмов:**
• Алгоритм легко понять. • Алгоритм — это пошаговое представление решения данной задачи. • В алгоритме проблема разбивается на более мелкие части или шаги, поэтому программисту легче преобразовать ее в настоящую программу.	• Написание алгоритма занимает много времени, поэтому оно отнимает много времени. • Понимание сложной логики с помощью алгоритмов может быть очень трудным. • Операторы ветвления и цикла трудно показать в алгоритме.

Что такое структура данных?

Структура данных — это математическая или логическая модель организации данных. Короче говоря, структура данных — это способ организации данных в форме, доступной для компьютеров. Он позволяет обрабатывать большие объемы данных за относительно короткий промежуток времени. Основная цель использования структур данных — сократить временные и пространственные сложности. Эффективная структура данных использует минимум памяти и требует минимально возможного времени для выполнения.

Пример и классная работа

Алгоритм сложения 3 чисел и вывода их суммы:

- Получить от пользователя 3 целочисленные переменные num1, num2 и num3.
- Возьмите три добавляемых числа в качестве входных данных для переменных num1, num2 и num3 соответственно.
- Объявите целочисленную переменную sum для хранения результирующей суммы трех чисел.
- Добавьте 3 числа и сохраните результат в переменной sum.
- Вывести значение переменной sum

Задача будет считаться выполненной если:

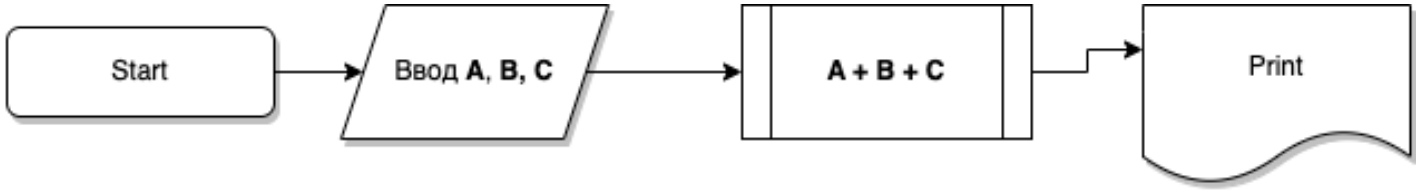
1. Написан псевдокод, диаграмма или написан код на JAVA or JavaScript
2. Алгоритм соответствует характеристикам алгоритма и выдает верный результат

Решения в следующем слайде 👇

Решения в 3х видах

Псевдокод	JAVA	Java Script

START
get number input: a, b, c
declare sum
sum = a+b+c
print sum
END
	public static void main(String[] args)
	{
	int sum = 0;
	Scanner sc = new Scanner(System.in);
	
	System.out.println("Enter the 1-st number: ");
	int num1 = sc.nextInt();
	
	System.out.println("Enter the 2-nd number: ");
	int num2 = sc.nextInt();
	;
	System.out.println("Enter the 3-rd number: ");
	int num3 = sc.nextInt();
	
	sum = num1 + num2 + num3;
	
	System.out.println("Sum of the 3 numbers is = " + sum);
	}
		function threeNumbersSum() {
		let sum = 0;
		let num1 = parseInt(prompt("Enter the 1st number: "));
		console.log(` ${num1}`);
		let num2 = parseInt(prompt("Enter the 2nd number: "));
		console.log(` ${num2}`);
		let num3 = parseInt(prompt("Enter the 3rd number: "));
		console.log(` ${num3}`);
		sum = num1 + num2 + num3;
		console.log(`Sum of the 3 numbers is = ${sum}`);
		}
		
		threeNumbersSum();



Домашнее задание

Level 1

Найти индекс заданного числа в массиве: `{3, 6, 4, 7, 2, 1, 9}`
Алгоритм на вход должен получать любой массив и одну цифру, индекс которой требуется найти.

Задание считается выполненным если: использован ”Линейный подход” и алгоритм соответствует характеристикам алгоритма и выдает верный результат.

Level 2

Реализовать алгоритм, который будет находить сумму квадратов всех элементов массива! `{3, 6, 4, 7, 2, 1, 9}`

Задание считается выполненным если: использован ”Линейный подход” и алгоритм соответствует характеристикам алгоритма и выдает верный результат.