Algorithms and data structures

lecture #1Introduction, simple examples.

Mentor: <….>

lecture #1 Introduction, simple examples.

- What is algorithm, introduction
- Characteristics of an algorithm
- Properties of an algorithm
- Type of an algorithm
- How to design an algorithm
- Example
  - add three numbers and print the sum
- Why analysis is important
  - Algorithmic analysis
  - Types of algorithmic analysis

What is algorithm, introduction

- **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

Characteristics of an algorithm

- **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.

Properties of an algorithm

- 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.

Type of an algorithm

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.

How to design an algorithm

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.

Пример

Существует три основных способа описания алгоритма:

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

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

Фактор времени : время измеряется путем подсчета количества ключевых операций.

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

Space Complexity

Time Complexity

Преимущества 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.

Недостатки алгоритмов:

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.