Лекция 1. Введение в анализ алгоритмов. Роль алгоритмов в вычислениях

Казахский национальный университет имени аль-Фараби

Кафедра компьютерных наук

Алгоритмы и структуры данных

Лектор: старший преподаватель, PhD, Кенжебек Ержан Ғалымжанұлы

Цель и план лекции

Цель лекции

Сформировать представление об алгоритме как о корректной вычислительной процедуре и о его роли в решении практических задач; научиться формулировать вычислительную задачу и проверять корректность алгоритма.

План лекции

1. Понятие алгоритма и вычислительной задачи
2. Пример: задача сортировки
3. Корректность алгоритма
4. Задачи, решаемые с помощью алгоритмов
5. Линейный и бинарный поиск

Что такое алгоритмы?

- Что такое алгоритм?
  - Это процедура выполнения определенной задачи. Алгоритм является основополагающей идеей любой компьютерной программы. Чтобы представлять интерес, алгоритм должен решать общую, корректно поставленную задачу. Определение задачи, решаемой с помощью алгоритма, дается описанием всего множества экземпляров, которые должен обработать алгоритм, и выхода, т. е. результата, получаемого после обработки одного из этих экземпляров.
  - Алгоритм — это любая четко определенная вычислительная процедура, которая берет некоторое значение или набор значений в качестве входных данных и производит некоторое значение или набор значений в качестве выходных. Алгоритм, таким образом, представляет собой последовательность вычислительных шагов, которые преобразуют ввод в вывод

Пример

- **ЗАДАЧА. Сортировка.**

Вход. Последовательность из n элементов: A1,A2, ..., An.

Выход. Перестановка элементов входной последовательности таким образом, что для ее членов справедливо соотношение

$$A1 <= A2 <= \ldots <= An$$

Например, учитывая входную последовательность: {31, 41, 59, 26, 41, 58}, алгоритм сортировки возвращает на выходе последовательность{26, 31, 41, 41, 58, 59}.

Такая входная последовательность называется экземпляром задачи сортировки. В общем, пример проблемы состоит из входных данных (удовлетворяющих любым ограничениям, наложенным в задаче оператор), необходимый для вычисления решения задачи.

Сортировка является основополагающей операцией в информатике, **в результате чего появилась большое количество хороших алгоритмов.**

Корректность алгоритма

- Говорят, алгоритм корректен, если для любых входных данных результатом его работы будет корректные выходные данные.
- Корректный алгоритм решает данную вычислительную задачу. Если алгоритм некорректный, то для некоторых вводов он может вообще не завершить свою работу

Задачи и свойства

- Чтобы продемонстрировать правильность алгоритма, одного его описания недостаточно. Нам также нужно подробное описание задачи, которая подлежит решению.
- Постановка задачи состоит из двух частей: **набора допустимых входных экземпляров и требований к выходу алгоритма**. Невозможно доказать правильность алгоритма для нечетко поставленной задачи. Иными словами, поставьте неправильно задачу, и вы получите неправильный ответ.
- При указании требований выхода алгоритма часто допускаются две ошибки.
  - Первая из них— плохая формулировка вопроса. Примером может служить вопрос о наилучшем маршруте между двумя точками на карте при отсутствии определения, что значит "**наилучший**". Лучший в каком смысле? **В смысле самого короткого расстояния, самого короткого времени прохождения или, может быть, минимального количества поворотов**?
  - Второй ошибкой является формулирование составных целей. Каждый из трех только что упомянутых критериев наилучшего маршрута является четко определенной целью правильного эффективного алгоритма оптимизации. (то есть задача мин. поворота или оптимальная прохождения по времени)

Демонстрация неправильности алгоритма

- Самый лучший способ доказать **неправильность** алгоритма — найти **экземпляр задачи (instance of the problem)**, для которого алгоритм выдает неправильный ответ. Такие экземпляры задачи называются **контрпримерами**. Хороший контрпример должен обладать двумя важными свойствами:
  - проверяемостью. Чтобы продемонстрировать, что некий входной экземпляр задачи является контрпримером для определенного алгоритма, требуется вычислить ответ, который алгоритм выдаст для данного экземпляра, и предоставить лучший ответ, с тем, чтобы доказать, что алгоритм не смог его найти;
  - простотой. Хороший контрпример не содержит ничего лишнего и ясно демонстрирует, почему именно предлагаемый алгоритм неправильный. Поэтому обнаруженный контрпример следует упростить.

Демонстрация неправильности алгоритма

- Развитие навыков поиска, контрпримеров будет стоить затраченного времени. Этот процесс в чем-то подобен разработке наборов тестов для проверки компьютерных программ, но в нем главную роль играет удачная догадка, а не перебор вариантов. Приведу несколько советов:
  - Ищите мелкомасштабные решения. **(найти подходящий простой экземпляр)**
  - Рассмотрите все решения (**рассмотрите все случаи для небольшого количества экземпляров**)
  - Ищите слабое звено. Если рассматриваемый вами алгоритм работает по принципу "всегда берем самое большее" (так называемый жадный алгоритм), то подумайте, почему этот подход может быть неправильным.
  - Ищите ограничения. "Жадный" эвристический алгоритм можно сбить с толку необычным методом предоставления входного экземпляра, содержащего одинаковые элементы. В этой ситуации алгоритму будет не на чем основывать свое решение и он, возможно, возвратит неоптимальное решение
  - Рассматривайте крайние случаи. Многие контрпримеры представляют собой комбинацию большого и малого, правого и левого, многих и немногих, далекого близкого. Обычно легче осмыслить и проверить примеры по краям диапазона, чем из его середины.

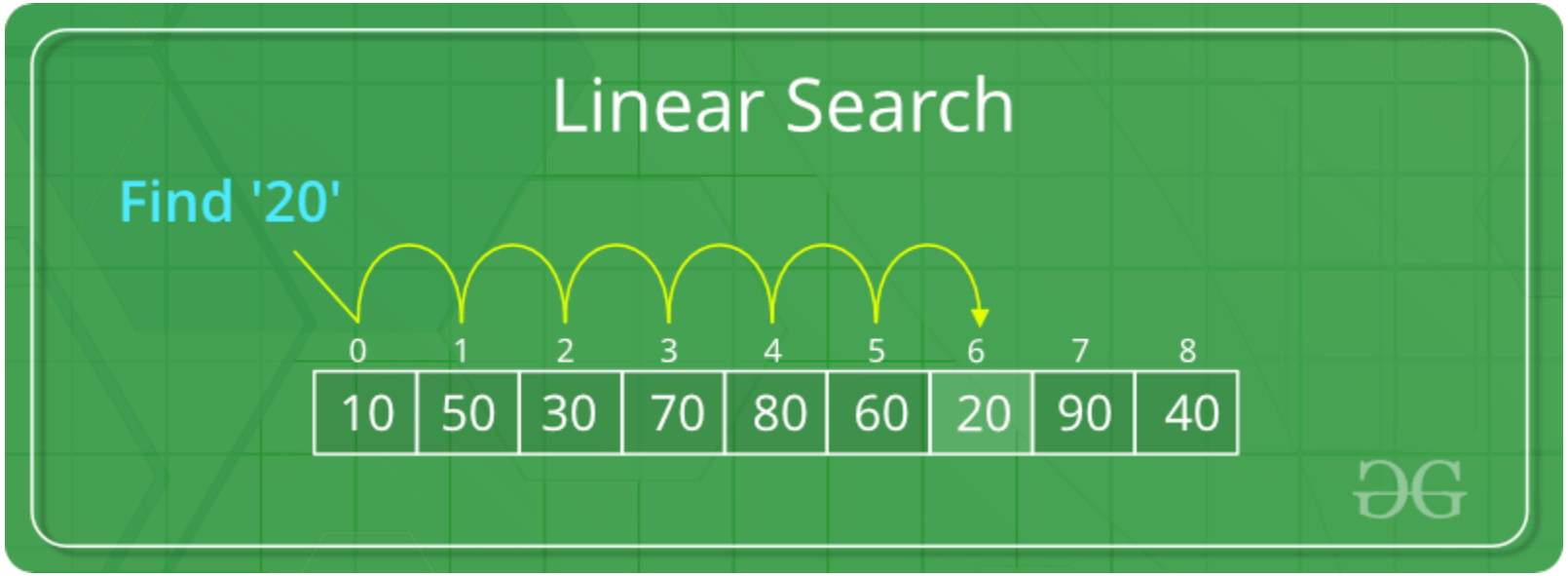
Какие задачи решаюся с помощью алгоритмов?

- Практические приложения алгоритмов повсеместны и включают следующие примеры:
  - Проект «Геном человека» значительно продвинулся к цели идентификации всех 100 000 генов в ДНК человека, определения последовательностей ДНК. 3 миллиарда пар химических оснований, из которых состоит ДНК человека, хранение этой информации в базах данных и разработка инструментов для анализа данных. Каждый из этих шагов требует сложных алгоритмов.
  - Электронная коммерция позволяет вести переговоры и обменивать товары и услуги в электронном виде, и это зависит от конфиденциальности личной информации, такой как номера кредитных карт, пароли и банковские выписки. (**Cryprography**)
  - Производственным и другим коммерческим предприятиям часто приходится выделять дефицитные ресурсы наиболее выгодным образом. Нефтяная компания может пожелать знать, где разместить свои скважины так, чтобы максимизировать ожидаемую прибыль.
  - В логистике, в ГИС приложениях и везде где требуется определенная последовательность действий (алгоритм)
  - И другие примеры.

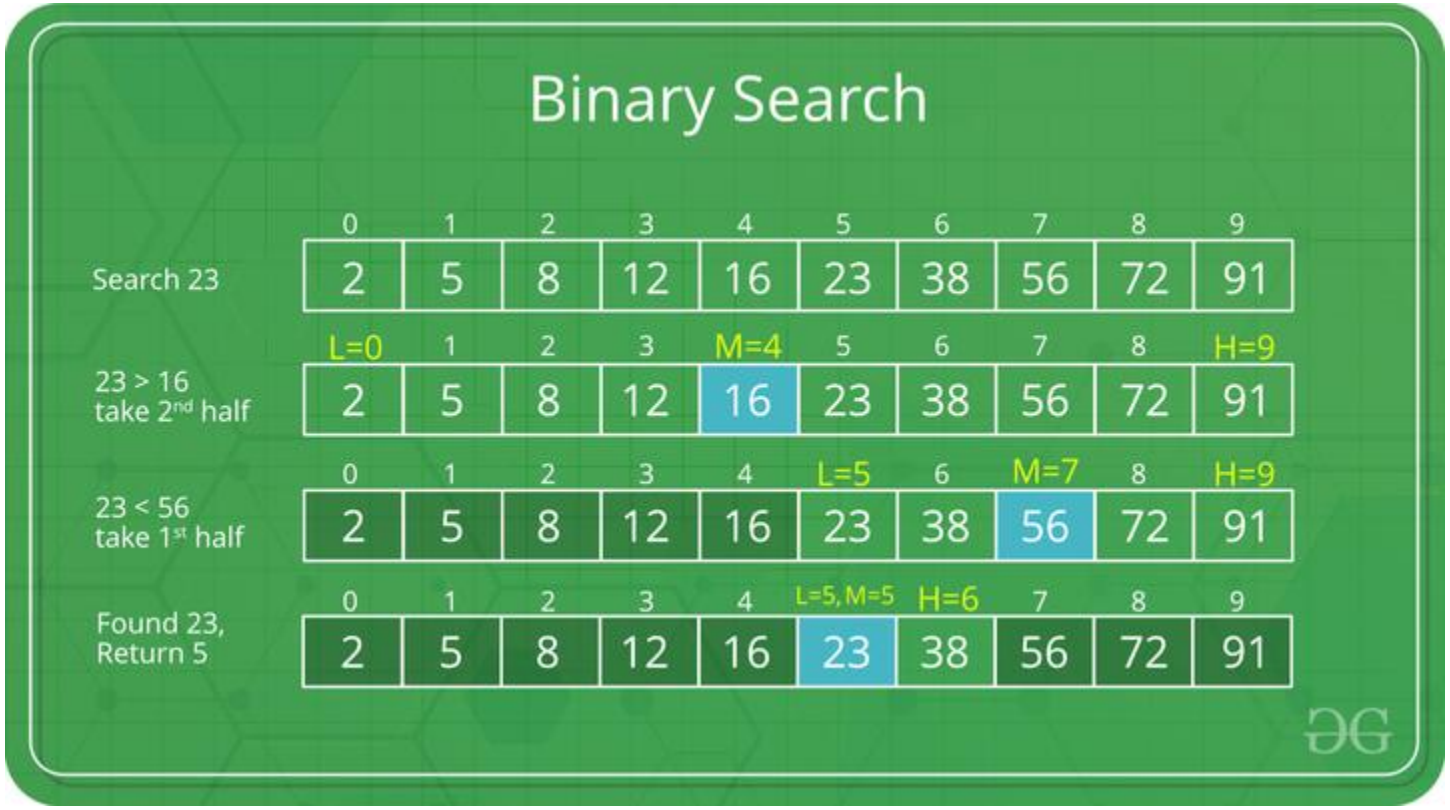
Пример алгоритмов

- Линейный и бинарный поиск.

Linear Search



Binary Search



Выводы

1. Алгоритм — чётко определённая вычислительная процедура, преобразующая входные данные в выходные.
2. Постановка задачи включает множество допустимых входов и требования к выходу; цели должны быть сформулированы чётко и не быть составными.
3. Алгоритм корректен, если для любого допустимого входа он завершается с правильным результатом; некорректность доказывается контрпримером.
4. Бинарный поиск эффективнее линейного, но применим только к отсортированным данным.

Контрольные вопросы

1. Дайте определение алгоритма. Чем алгоритм отличается от программы?
2. Из каких частей состоит постановка вычислительной задачи?
3. Какой алгоритм называется корректным? Как показать, что алгоритм некорректен?
4. Какие ошибки часто допускают при формулировке требований к результату алгоритма?
5. Опишите принцип работы бинарного поиска и условие его применимости.
6. Сравните линейный и бинарный поиск по числу сравнений в худшем случае.

Список литературы

Основная литература

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. 3-е изд. — М.: Вильямс, 2013.
2. Скиена С. Алгоритмы. Руководство по разработке. 2-е изд. — СПб.: БХВ-Петербург, 2011.
3. Левитин А. Алгоритмы: введение в разработку и анализ. — М.: Вильямс, 2006.

Дополнительная литература

4. Бхаргава А. Грокаем алгоритмы. — СПб.: Питер, 2017.
5. Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы. — М.: МЦНМО, 2014.
6. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms. 4th ed. — MIT Press, 2022.

Электронные ресурсы

7. MIT OpenCourseWare, курс 6.006 «Introduction to Algorithms»: ocw.mit.edu