Университет Сириус
Колледж

Алгоритмы, программирование и структуры данных
Лекция 1. Введение

Введение. Алгоритм. Память и время как ресурсы. О-символика как инструмент оценки ресурсов, асимптотики. Метод математической индукции

Университет Сириус
Колледж

Алгоритм

это набор шагов или инструкций, необходимых для выполнения задачи.

Алгоритм

это набор конечного числа правил, задающих последовательность выполнения операций компьютерной программой для решения задачи определённого типа.

Университет Сириус Колледж

Зачем изучать алгоритмы?

- Существует устоявшийся массив знаний о том, как хорошо решать конкретные проблемы, и важно знать, какие решения есть сейчас.
- Необходимо знать не только о существовании подходящего алгоритма, но и понимать когда его применить, определять тип задачи.

Университет Сириус
Колледж

Алгоритмическое мышление

Университет Сириус Колледж

Свойства алгоритма

- **Конечность.** Алгоритм всегда должен заканчиваться после выполнения конечного числа шагов.
- **Определённость.** Действия, которые нужно выполнить, должны быть строго и недвусмысленно определены для каждого возможного случая.
- **Вввод.** Алгоритм имеет некоторое (возможно, равное нулю) число входных данных.
- **Вывод.** У алгоритма есть одно или несколько выходных данных , т. е. величин, имеющих вполне определенную связь с входными данными.
- **Эффективность.** Алгоритм обычно считается эффективным, если все его операторы достаточно просты для того, чтобы их можно было точно выполнить в течение конечного промежутка времени с помощью карандаша и бумаги.

Университет Сириус Колледж

Алгоритм E (Алгоритм Евклида).

Даны два целых положительных числа m и n. Требуется найти их наибольший общий делитель, т.е. наибольшее целое положительное число, которое нацело делит оба числа m и n.

1. Нахождение остатка. Разделим m на n, и пусть остаток от деления будет равен r (где $0 \leq r < n$).

2. Сравнение с нулем. Если $r = 0$, то выполнение алгоритма прекращается; n — искомое значение.

3. Замещение. Присвоить $m = n, n = r$ и вернуться к шагу 1.

Университет Сириус
Колледж

Память и время

Университет Сириус Колледж

Память

Использование основной памяти (зачастую, RAM) нужной алгоритму.

Для анализа алгоритма обычно используется анализ пространственной сложности алгоритма, чтобы оценить необходимую память времени исполнения как функцию от размера входа. Результат обычно выражается в терминах **«О» большое**.

Университет Сириус Колледж

Память

Существует четыре аспекта использования памяти:

- Количество памяти, необходимой для хранения кода алгоритма.
- Количество памяти, необходимое для входных данных.
- Количество памяти, необходимое для любых выходных данных (некоторые алгоритмы, такие как сортировки, часто переставляют входные данные и не требуют дополнительной памяти для выходных данных).
- Количество памяти, необходимое для вычислительного процесса во время вычислений (сюда входят именованные переменные и любое стековое пространство, необходимое для вызова подпрограмм, которое может быть существенным при использовании рекурсии).

Университет Сириус Колледж

Время

Для анализа алгоритма обычно используется анализ временной сложности алгоритма, чтобы оценить время работы как функцию от размера входных данных. Результат обычно выражается в терминах «О» большое

Университет Сириус
Колледж

О-символика как инструмент оценки ресурсов, асимптотики

Университет Сириус Колледж

«O» символика

«Analytische Zahlentheorie» (Аналитическая теория чисел), 1894 год

Обозначение от немецкого слова «Ordnung» (порядок)

Пауль Бахман

Университет Сириус
Колледж

«O» большое

математическое обозначение для сравнения асимптотического поведения (асимптотики) функций.

Используются в различных разделах математики, но активнее всего — в математическом анализе, теории чисел и комбинаторике, а также в информатике и теории алгоритмов. Под **асимптотикой** понимается характер изменения функции при стремлении её аргумента к определённой точке.

«O» символика

Запись $O(f(n))$ всегда обозначает следующее: существуют положительные константы M и n_0 , такие, что величина x_n, представленная в виде $O(f(n))$, удовлетворяет условию $|x_n| \leq M|f(n)|$ для всех целых $n \geq n_0$.

Университет Сириус
Колледж

Фраза **«сложность алгоритма есть $O(f(n))$»** означает, что с увеличением параметра n, характеризующего количество входной информации алгоритма, время работы алгоритма будет возрастать не быстрее, чем $f(n)$, умноженная на некоторую константу.

Университет Сириус
Колледж

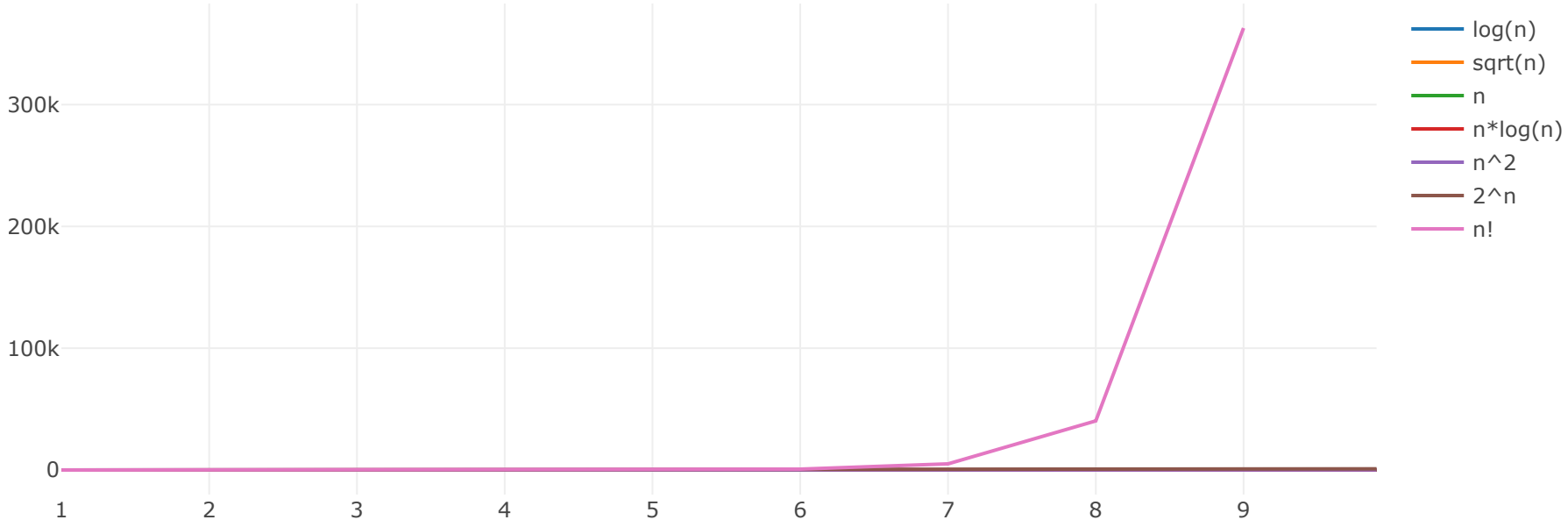
Операции над символом O

$$f(n) = O(f(n))$$
$$c \cdot O(f(n)) = O(f(n))$$
$$O(f(n)) + O(f(n)) = O(f(n))$$
$$O(O(f(n))) = O(f(n))$$
$$O(f(n))O(g(n)) = O(f(n)g(n))$$
$$O(f(n)g(n)) = f(n)O(g(n))$$

Университет Сириус Колледж

Асимптотики

$$\log n \prec \sqrt{n} \prec n \prec n \log n \prec n^2 \prec 2^n \prec n!$$



Университет Сириус
Колледж

Метод математической индукции

Университет Сириус
Колледж

Метод математической индукции

1. Доказать, что P(1) верно.
2. Доказать, что “если P(1), P(2), . . . , P(n) справедливы, то P(n + 1) также справедливо”; это доказательство должно иметь силу для любого целого положительного n.

Университет Сириус Колледж

Пример

$$1 = 1^2$$

$$1 + 3 = 2^2$$

$$1 + 3 + 5 = 3^2$$

$$1 + 3 + 5 + 7 = 4^2$$

$$1 + 3 + 5 + 7 + 9 = 5^2$$

$$\dots$$

$$1 + 3 + \dots + (2n - 1) = n^2$$

$$P(n+1) : 1 + 3 + \dots + (2n - 1) + (2n + 1) = (n + 1)^2$$

Университет Сириус Колледж

Алгоритм B (двоичный (бинарный) поиск)

Дана таблица записей $R_1, R_2, \ldots, R_N$, ключи которых расположены в порядке возрастания: $K_1 < K_2 < \ldots < K_N$; алгоритм используется для поиска в таблице заданного аргумента K.

1. Установить $l = 1, u = N$.
2. Если $u < l$, алгоритм завершается неудачно; иначе установить $i = floor((l + u)/2)$, чтобы i соответствовало примерно середине рассматриваемой части таблицы.
3. Если $K < K_i$, перейти к шагу 4; если $K > K_i$, перейти к шагу 5; если $K == K_i$, алгоритм успешно завершается.
4. Установить $u = i - 1$ и перейти к шагу 2.
5. Установить $l = i + 1$ и перейти к шагу 2.

Университет Сириус Колледж

Мат. индукция для алгоритма B

$$n = 1 : \ 2^0 + 1 \ : \ 1 \text{ шаг}$$

$$n = 2 : \ 2^1 + 1 \ : \ 1 - 2 \text{ шага}$$

$$n = 3 : \ 2^1 + 1 \ : \ 2 \text{ шага}$$

$$n = 4 : \ 2^2 + 1 \ : \ 3 \text{ шага}$$

$$n = 8 : \ 2^3 + 1 \ : \ 4 \text{ шага}$$

$$n = 16 : \ 2^4 + 1 \ : \ 5 \text{ шагов}$$

$$n = 32 : \ 2^5 + 1 \ : \ 6 \text{ шагов}$$

$$n : log(n) + 1 \text{ шагов} \Rightarrow O(n) = log(n)$$