

Меню



Алгоритмы и Структуры данных. Вычислительная сложность

Процесс



Зачем это учить?

Вспоминаем и обобщаем.

[← к предыдущему разделу](#)

[к следующей теме >](#)

Как это учить?

Нужно внимательно прочитать и понять материал темы и выполнить задания. Задания должны быть загружены в Булгаков в течение недели после прохождения темы. Решение задач оформляется в виде программного кода, расчет времени выполнения записывается в нотации "О-большое".

Материал, предложенный к ознакомлению по ссылке носит справочный характер и является факультативным.

Структура курса и КТ



1

Повторение



3

АиСД. Продолжение?

3 курс 1 семестр – ЗП1, ЗП2
2 часа в неделю (всего **28 часов**)

Каждое занятие – задача, чтоб
было нескучно 😊

2

Темы, оставшиеся с прошлого года:

1. Кратчайшие пути в графах:
 - a. алгоритм Беллмана-Форда
 - b. алгоритм Флойда-Уоршалла
2. Шифрование:
 - a. алгоритм Диффи-Хеллмана
 - b. алгоритм RSA



Структура КТ

ithub.ru

1

Повторение

- ✓ ДЗ оценивается!
- ✓ 5 баллов каждое
- ✓ но необязательно к выполнению

3

АиСД. Продолжение?

3 курс 1 семестр
2 часа в неделю (всего **28 часов**)

2

Темы, оставшиеся с прошлого года:

1. Кратчайшие пути в графах:
 - ✓ КТ на применение алгоритмов
 - ✓ 15 баллов
2. Шифрование:
 - ✓ КТ теория
 - ✓ 15 баллов

Задача “чтоб нескучно”

- ✓ Задача целиком 30 баллов
- ✓ Идея 5 баллов

Для тех, кому все и сразу и в самом конце:

- ✓ 2 большие работы по 50 баллов.
- ✓ Но не рекомендую!!!



- ☐ Массив, сортированный массив
- ☐ Связные списки
- ☐ Стек, очередь, дек
- ☐ Хеш-таблицы
- ☐ Бинарные деревья
- ☐ Куча

- ☐ Сравнением (пузырек, вставками, выбором)
- ☐ Слиянием
- ☐ Быстрая
- ☐ Кучей
- ☐ Подсчетом, поразрядная

Дизайн алгоритмов

- ☐ Рекурсия
- ☐ Разделяй и властвуй
- ☐ Жадные алгоритмы
- ☐ Динамическое программирование

Кратчайшие пути

- ☐ Обход в глубину
- ☐ Обход в ширину
- ☐ Дейкстра
- ☐ Топологическая сортировка
- ☐ Минимальное остовное дерево

Задача и алгоритм

Задача:

- Имеет вход (input) и выход (output),
- Каждому входу соответствует "правильный" выход
- Есть некое словесное описание "правильности" выхода

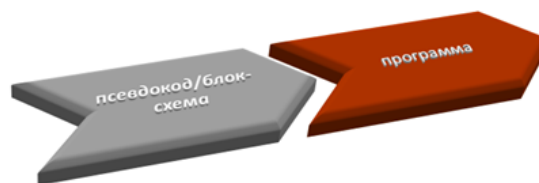
Для алгоритмических задач **пространство входов произвольно большое**.

Пример

- Есть ли среди студентов ЗП1 те, у кого др в один день?
- vs
- Есть ли среди n студентов те, у кого др в один день?

Алгоритм:

- это процедура
- Каждому входу сопоставляет один выход (детерминированность)
- Алгоритм "решает" задачу, если для каждого входа выдает "правильный" выход (массовость + конечность + полезность)



Вычислительная сложность: считаем число операций (выполняющихся за фиксированное время) как функцию от размера входа.

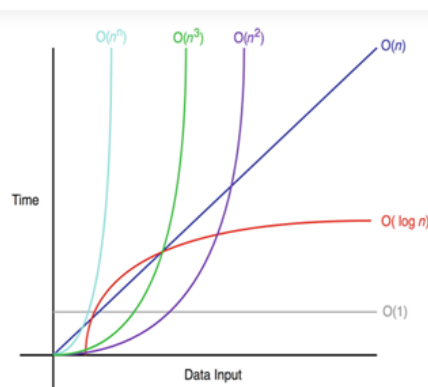


$O(n)$ – *асимптотическая верхняя* граница сложности алгоритма

Асимптотическая значит, что *игнорируем постоянные множители и слагаемые меньшего порядка*

Другие нотации:

- ✓ $\Omega(n)$ нижняя граница сложности
- ✓ $\Theta(n)$ точная оценка сложности



10^4	2,8 часа
10^5	1,1 дня
10^6	1,6 недели
10^7	3,8 месяца
10^8	3,1 года
10^9	3,1 десятилетия
10^{10}	3,1 столетия
10^{11}	никогда

Частиц во вселенной $< 10^{100}$

input	constant	logarithmic	linear	log-linear	quadratic	polynomial	exponential
n	$\Theta(1)$	$\Theta(\log n)$	$\Theta(n)$	$\Theta(n \log n)$	$\Theta(n^2)$	$\Theta(n^c)$	$2^{\Theta(n^c)}$
1000	1	≈ 10	1000	$\approx 10,000$	1,000,000	1000^c	$2^{1000} \approx 10^{301}$
Time	1 ns	10 ns	1 μ s	10 μ s	1 ms	10^{k-9} s	10^{281} millennia

Время приведено для 1 GHz одноядерного компа

Структуры данных

Структура данных = объекты + интерфейс,

т.е. способ хранения данных, который поддерживает выполнение определенных операции над ними

Абстрактная структура данных – один и тот же интерфейс, но возможна его различная реализация

Последовательность - ?
Множество - ?

Для различных структур данных
вычислительная сложность операций
вставка, удаление и поиск разная !

Наивная реализация vs Обоснование корректности

Идея: можно реализовать "очевидный" алгоритм, а можно поискать способы более эффективные:

- либо за счет выбора структуры данных
- либо за счет изменения логики алгоритма

При "неочевидной" логике алгоритма, необходимо обосновать, что алгоритм работает правильно. При выборе структуры данных надо понять, значительно ли изменится вычислительная сложность.

[< к предыдущему разделу](#)

[к следующей теме >](#)

Задания

#1

ДЗ 1. Каждый вопрос 5 баллов

Ваш ответ на это задание увидят только преподаватели и вы

1. Есть ли совпадающие дни рождения в группе?

1. Напишите программу, которая будет отвечать на вопрос: "Есть ли среди n студентов те, у кого день рождения в один день?".

Колонин Глеб



[Показать больше...](#)

Ответ

Параграф



Добавить вложения



2. Асимптотическая оценка

1. Придумайте 10 функций $f(n)$, напишите оценку $O(n)$ для них и расположите их в порядке убывания вычислительной сложности (сначала более быстрые, потом менее быстрые)
2. Посчитайте $O(n)$ для

$$\log_{6006} \left(\left(\log \left(n^{\sqrt{n}} \right) \right)^2 \right).$$

3. Приведите примеры алгоритмов с вычислительной сложностью $O(1)$, $O(\log n)$, $O(n \log n)$, $O(n)$, $O(n^2)$, $O(2^n)$.

[Показать меньше...](#)

Ответ

Параграф



Добавить вложения



Отправить

Мои ответы

Вы еще не отвечали на задания этой темы