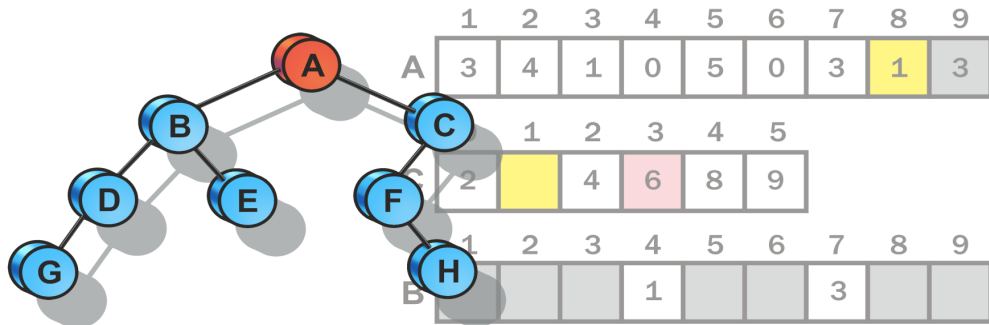
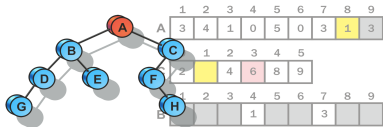


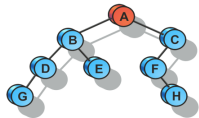
Алгоритмы программирования и структуры данных



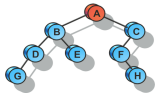


Алгоритмы программирования и структуры данных

Введение



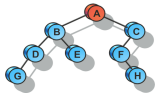
**Алгоритм. Структура данных. Алгоритмы и
структуры данных как эффективная технология**



Алгоритмы программирования и структуры данных

О чем этот курс?

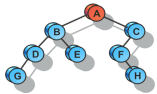
- ▶ Алгоритмы



Алгоритмы программирования и структуры данных

О чем этот курс?

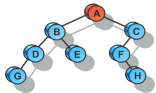
- ▶ Алгоритмы
- ▶ Структуры данных



Алгоритмы программирования и структуры данных

О чем этот курс?

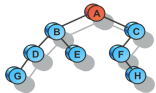
- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных



Алгоритмы программирования и структуры данных

О чем этот курс?

- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных
- ▶ Реализация, оценка времени работы и используемой памяти

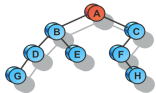


Алгоритмы программирования и структуры данных

О чем этот курс?

- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных
- ▶ Реализация, оценка времени работы и используемой памяти

Более подробно



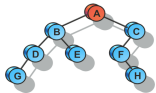
Алгоритмы программирования и структуры данных

О чем этот курс?

- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных
- ▶ Реализация, оценка времени работы и используемой памяти

Более подробно

- ▶ Алгоритмы сортировки



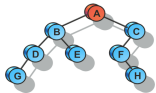
Алгоритмы программирования и структуры данных

О чем этот курс?

- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных
- ▶ Реализация, оценка времени работы и используемой памяти

Более подробно

- ▶ Алгоритмы сортировки
- ▶ Линейные структуры данных: списки, стеки, очереди



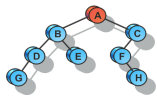
Алгоритмы программирования и структуры данных

О чем этот курс?

- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных
- ▶ Реализация, оценка времени работы и используемой памяти

Более подробно

- ▶ Алгоритмы сортировки
- ▶ Линейные структуры данных: списки, стеки, очереди
- ▶ Алгоритмы и структуры данных для эффективного хранения и поиска информации: сбалансированные деревья поиска, хеш-таблицы



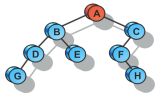
Алгоритмы программирования и структуры данных

О чем этот курс?

- ▶ Эффективные алгоритмы
- ▶ Эффективные структуры данных
- ▶ Реализация, оценка времени работы и используемой памяти

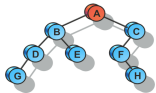
Более подробно

- ▶ Алгоритмы сортировки
- ▶ Линейные структуры данных: списки, стеки, очереди
- ▶ Алгоритмы и структуры данных для эффективного хранения и поиска информации: сбалансированные деревья поиска, хеш-таблицы
- ▶ Строковые алгоритмы



Что такое алгоритм?

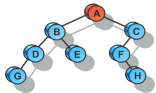
Алгоритм — корректно определенная вычислительная процедура, принимающая на вход и выдающая на выход данные определенного вида.



Что такое алгоритм?

Алгоритм — корректно определенная **вычислительная процедура**, принимающая на вход и выдающая на выход данные определенного вида.

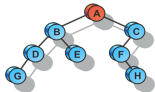
Вычислительная процедура — последовательность формальных манипуляций с данными.



Что такое алгоритм?

Алгоритм — корректно определенная вычислительная процедура, принимающая на **вход** и выдающая на выход **данные** определенного вида.

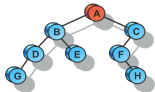
Входные данные — данные в определенном формате, с которыми будет работать алгоритм.



Что такое алгоритм?

Алгоритм — корректно определенная вычислительная процедура, принимающая на вход и выдающая на **выход данные** определенного вида.

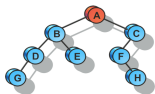
Выходные данные — данные в определенном формате, которые алгоритм вернет после окончания работы.



Что такое алгоритм?

Алгоритм — корректно определенная вычислительная процедура, принимающая на вход и выдающая на выход данные определенного вида.

Корректный алгоритм для любых допустимых входных данных выполняется конечное время и выдает корректные выходные данные.

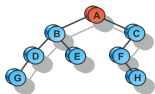


Что такое алгоритм?

Алгоритм — корректно определенная вычислительная процедура, принимающая на вход и выдающая на выход данные определенного вида.

Корректный алгоритм для любых допустимых входных данных выполняется за конечное время и выдает **корректные выходные данные**.

Корректность выходных данных: алгоритм, как правило, решает некоторую задачу. Выходные данные должны удовлетворять требованиям, поставленным в задаче.



Что такое алгоритм?

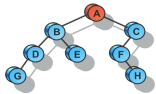
Алгоритм — корректно определенная вычислительная процедура, принимающая на вход и выдающая на выход данные определенного вида.

Корректный алгоритм для любых допустимых входных данных выполняется за конечное время и выдает корректные выходные данные.

Корректность выходных данных: алгоритм, как правило, решает некоторую **задачу**. Выходные данные должны удовлетворять требованиям, поставленным в задаче.

Вычислительная задача: описание множества входных данных и требований к выходным данным.

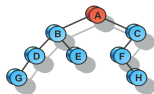
Экземпляр задачи: конкретные входные данные, удовлетворяющие описанию в задаче.



Пример алгоритма

Задача

- ▶ Отсортировать набор чисел
- ▶ Входные данные: набор целых чисел
- ▶ Выходные данные: набор этих же чисел, упорядоченный по неубыванию



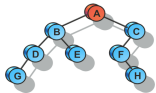
Пример алгоритма

Задача

- ▶ Отсортировать набор чисел
- ▶ Входные данные: набор целых чисел
- ▶ Выходные данные: набор этих же чисел, упорядоченный по неубыванию

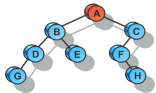
Алгоритм (сортировка вставками)

- ▶ Пусть дан набор чисел A , размер набора n
- ▶ Для всех i от 2 до n :
 - ▶ Пусть $j = i$
 - ▶ Пока $j > 1$ и $A_{j-1} > A_j$
 - ▶ $A_{j-1} \leftrightarrow A_j$
 - ▶ $j \leftarrow j - 1$



Что такое структура данных?

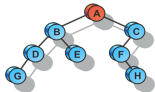
Структура данных — способ организации данных, корректно поддерживающий определенный набор операций.



Что такое структура данных?

Структура данных — способ организации данных, корректно поддерживающий определенный набор операций.

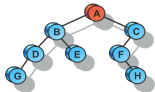
Данные хранятся **в определенном состоянии** и не изменяются, если с ними не производятся операции.



Что такое структура данных?

Структура данных — способ организации данных, **корректно** поддерживающий определенный **набор операций**.

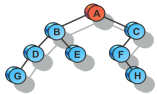
Операции с данными — алгоритмы, принимающие на вход имеющиеся данные и либо возвращающие определенную информацию об этих данных (**запросы**), либо изменяющие сами данные (**модификации**).



Что такое структура данных?

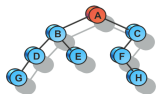
Структура данных — способ организации данных, корректно поддерживающий определенный набор операций.

Все операции должны оставлять структуру данных в **корректном состоянии** и, как правило, предполагается, что операции применяются к структуре данных, находящейся в **корректном состоянии**.



Пример структуры данных

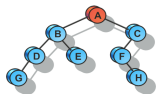
- ▶ Задача: поиск и добавление слов в словарь



Пример структуры данных

- ▶ Задача: поиск и добавление слов в словарь
- ▶ Пример неэффективной структуры данных.

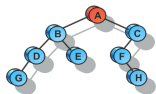
Рак Кабан Козел Лемминг Черепаха Муравьед Нутрия Индюк Ягуар Хамелеон
Коала Барсук Акула Соболь Осьминог Лошадь Крыса Зебра Кошка Кашалот
Крот Тюлень Индейка Верблюд Сурок Лисица Утка Заяц Бобр Суслик Аллигатор
Варан Леопард Игуана Овца Гиена Еж Ленивец Гадюка Крокодил Свинья Лось
Курица Кенгуру Дикобраз Хомяк Утконос Жираф Ондатра Бизон Скунс Обезьяна
Газель Койот Мышь Пингвин Осел Бурундук Тигр Олень Омар Пантера Пума
Кролик Ехидна Жаба Удав Кит Волк Морж Выхухоль Гепард Песец Енот
Антилопа Лемур Медведь Выдра Белка Гусь Гиппопотам Уж Рысь Зубр Куница
Динозавр Лев Кобра Собака Носорог Лягушка Хорек Корова Анаконда



Пример структуры данных

- ▶ Задача: поиск и добавление слов в словарь
- ▶ Пример неэффективной структуры данных. Найдите хомяка.

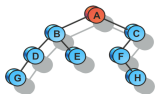
Рак Кабан Козел Лемминг Черепаха Муравьед Нутрия Индюк Ягуар Хамелеон
Коала Барсук Акула Соболь Осьминог Лошадь Крыса Зебра Кошка Кашалот
Крот Тюлень Индейка Верблюд Сурок Лисица Утка Заяц Бобр Суслик Аллигатор
Варан Леопард Игуана Овца Гиена Еж Ленивец Гадюка Крокодил Свинья Лось
Курица Кенгуру Дикобраз Хомяк Утконос Жираф Ондатра Бизон Скунс Обезьяна
Газель Койот Мышь Пингвин Осел Бурундук Тигр Олень Омар Пантера Пума
Кролик Ехидна Жаба Удав Кит Волк Морж Выхухоль Гепард Песец Енот
Антилопа Лемур Медведь Выдра Белка Гусь Гиппопотам Уж Рысь Зубр Куница
Динозавр Лев Кобра Собака Носорог Лягушка Хорек Корова Анаконда



Пример структуры данных

- ▶ Задача: поиск и добавление слов в словарь
- ▶ Пример неэффективной структуры данных. Найдите хомяка.

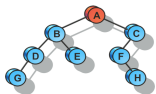
Рак Кабан Козел Лемминг Черепаха Муравьед Нутрия Индюк Ягуар Хамелеон
Коала Барсук Акула Соболь осьминог Лошадь Крыса Зебра Кошка Кашалот
Крот Тюлень Индейка Верблюд Сурок Лисица Утка Заяц Бобр Суслик Аллигатор
Варан Леопард Игуана Овца Гиена Еж Ленивец Гадюка Крокодил Свинья Лось
Курица Кенгуру Дикобраз **Хомяк** Утконос Жираф Ондатра Бизон Скунс Обезьяна
Газель Койот Мышь Пингвин Осел Бурундук Тигр Олень Омар Пантера Пума
Кролик Ехидна Жаба Удав Кит Волк Морж Выхухоль Гепард Песец Енот
Антилопа Лемур Медведь Выдра Белка Гусь Гиппопотам Уж Рысь Зубр Куница
Динозавр Лев Кобра Собака Носорог Лягушка Хорек Корова Анаконда



Пример структуры данных

- ▶ Задача: поиск и добавление слов в словарь
- ▶ Пример эффективной структуры данных.

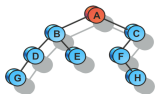
A: Акула Аллигатор Анаконда Антилопа **B:** Барсук Белка Бизон Бобр Бурундук
V: Варан Верблюд Волк Выдра Выхухоль **G:** Гадюка Газель Гепард Гиена
Гиппопотам Гусь **D:** Дикобраз Динозавр **E:** Енот Ехидна Еж **J:** Жаба Жираф **3:**
Заяц Зебра Зубр **I:** Игуана Индейка Индюк **K:** Кабан Кашалот Кенгуру Кит Кобра
Коала Козел Койот Корова Кошка Крокодил Кролик Крот Крыса Куница Курица **L:**
Лев Лемминг Лемур Ленивец Леопард Лисица Лось Лошадь Лягушка **M:** Медведь
Морж Муравьед Мышь **N:** Носорог Нутрия **O:** Обезьяна Овца Олень Омар
Ондатра Осел осьминог **P:** Пантера Песец Пингвин Пума **R:** Рак Рысь **S:** Свинья
Скунс Собака Соболь Сурок Суслик **T:** Тигр Тюлень **Y:** Удав Уж Утка Утконос **X:**
Хамелеон Хомяк Хорек **C:** Черепаха **Y:** Ягуар



Пример структуры данных

- ▶ Задача: поиск и добавление слов в словарь
- ▶ Пример эффективной структуры данных. **Найдите хомяка.**

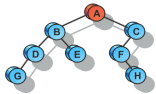
A: Акула Аллигатор Анаконда Антилопа **B:** Барсук Белка Бизон Бобр Бурундук
V: Варан Верблюд Волк Выдра Выхухоль **G:** Гадюка Газель Гепард Гиена
Гиппопотам Гусь **D:** Дикобраз Динозавр **E:** Енот Ехидна Еж **J:** Жаба Жираф **3:**
Заяц Зебра Зубр **I:** Игуана Индейка Индюк **K:** Кабан Кашалот Кенгуру Кит Кобра
Коала Козел Койот Корова Кошка Крокодил Кролик Крот Крыса Куница Курица **L:**
Лев Лемминг Лемур Ленивец Леопард Лисица Лось Лошадь Лягушка **M:** Медведь
Морж Муравьед Мышь **H:** Носорог Нутрия **O:** Обезьяна Овца Олень Омар
Ондатра Осел Осьминог **P:** Пантера Песец Пингвин Пума **P:** Рак Рысь **C:** Свинья
Скунс Собака Соболь Сурок Суслик **T:** Тигр Тюлень **Y:** Удав Уж Утка Утконос **X:**
Хамелеон Хомяк Хорек **Ч:** Черепаха **Я:** Ягуар



Пример структуры данных

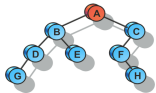
- ▶ Задача: поиск и добавление слов в словарь
- ▶ Пример эффективной структуры данных. **Найдите хомяка.**

A: Акула Аллигатор Анаконда Антилопа **B:** Барсук Белка Бизон Бобр Бурундук
V: Варан Верблюд Волк Выдра Выхухоль **G:** Гадюка Газель Гепард Гиена
Гиппопотам Гусь **D:** Дикобраз Динозавр **E:** Енот Ехидна Еж **Ж:** Жаба Жираф **З:**
Заяц Зебра Зубр **И:** Игуана Индейка Индюк **K:** Кабан Кашалот Кенгуру Кит Кобра
Коала Козел Койот Корова Кошка Крокодил Кролик Крот Крыса Куница Курица **L:**
Лев Лемминг Лемур Ленивец Леопард Лисица Лось Лошадь Лягушка **M:** Медведь
Морж Муравьед Мышь **H:** Носорог Нутрия **O:** Обезьяна Овца Олень Омар
Ондатра Осел Осьминог **P:** Пантера Песец Пингвин Пума **P:** Рак Рысь **C:** Свинья
Скунс Собака Соболь Сурок Суслик **T:** Тигр Тюлень **Y:** Удав Уж Утка Утконос **X:**
Хамелеон **Хомяк** Хорек **Ч:** Черепаха **Я:** Ягуар



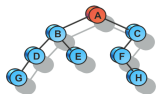
Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены



Эффективность алгоритмов

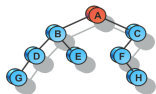
- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

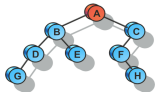
Вариант	1	2
---------	---	---



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

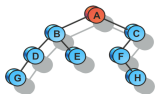
Вариант	1	2
Сортировка	Вставками	Слиянием



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

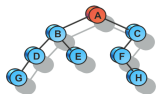
Вариант	1	2
Сортировка	Вставками	Слиянием
Сложность	$c_1 \cdot n^2$	$c_2 \cdot n \log_2 n$



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

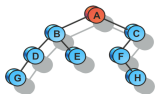
Вариант	1	2
Сортировка	Вставками	Слиянием
Сложность	$c_1 \cdot n^2$	$c_2 \cdot n \log_2 n$
Реализация	Ассемблер, $c_1 = 2$	Javascript, $c_2 = 100$



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

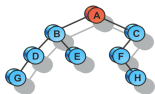
Вариант	1	2
Сортировка	Вставками	Слиянием
Сложность	$c_1 \cdot n^2$	$c_2 \cdot n \log_2 n$
Реализация	Ассемблер, $c_1 = 2$	Javascript, $c_2 = 100$
Вычислитель	Суперкомпьютер	Смартфон



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

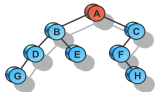
Вариант	1	2
Сортировка	Вставками	Слиянием
Сложность	$c_1 \cdot n^2$	$c_2 \cdot n \log_2 n$
Реализация	Ассемблер, $c_1 = 2$	Javascript, $c_2 = 100$
Вычислитель	Суперкомпьютер	Смартфон
Скорость	10^{11} операций/с	10^8 операций/с



Эффективность алгоритмов

- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

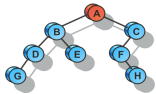
Вариант	1	2
Сортировка	Вставками	Слиянием
Сложность	$c_1 \cdot n^2$	$c_2 \cdot n \log_2 n$
Реализация	Ассемблер, $c_1 = 2$	Javascript, $c_2 = 100$
Вычислитель	Суперкомпьютер	Смартфон
Скорость	10^{11} операций/с	10^8 операций/с
Время, с	$\frac{2 \cdot 10^{16}}{10^{11}} = 2 \cdot 10^5$ с	$\frac{100 \cdot 10^8 \cdot 27}{10^8} = 2700$ с



Эффективность алгоритмов

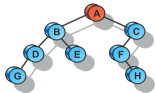
- ▶ Вычислительные ресурсы (время работы, память) всегда ограничены
- ▶ Используя эффективные алгоритмы, можно быстрее обрабатывать большие объемы данных
- ▶ Пример: сортировка массива размером $n = 10^8$ элементов

Вариант	1	2
Сортировка	Вставками	Слиянием
Сложность	$c_1 \cdot n^2$	$c_2 \cdot n \log_2 n$
Реализация	Ассемблер, $c_1 = 2$	Javascript, $c_2 = 100$
Вычислитель	Суперкомпьютер	Смартфон
Скорость	10^{11} операций/с	10^8 операций/с
Время, с	$\frac{2 \cdot 10^{16}}{10^{11}} = 2 \cdot 10^5$ с	$\frac{100 \cdot 10^8 \cdot 27}{10^8} = 2700$ с
Время	Более двух дней	44 минуты



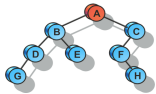
Зачем нужны алгоритмы и структуры данных?

- ▶ Интернет
 - ▶ маршрутизация пакетов (поиск кратчайших путей)
 - ▶ фильтрация спама (фильтры доменов)
 - ▶ функционирование веб-серверов (базы данных, строковые алгоритмы)



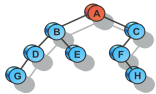
Зачем нужны алгоритмы и структуры данных?

- ▶ Интернет
 - ▶ маршрутизация пакетов (поиск кратчайших путей)
 - ▶ фильтрация спама (фильтры доменов)
 - ▶ функционирование веб-серверов (базы данных, строковые алгоритмы)
- ▶ Медицина и биология
 - ▶ Сборка генома (строковые алгоритмы, алгоритмы на графах, частотный анализ)
 - ▶ Разработка лекарств (вычислительная геометрия)



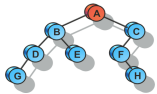
Зачем нужны алгоритмы и структуры данных?

- ▶ Интернет
 - ▶ маршрутизация пакетов (поиск кратчайших путей)
 - ▶ фильтрация спама (фильтры доменов)
 - ▶ функционирование веб-серверов (базы данных, строковые алгоритмы)
- ▶ Медицина и биология
 - ▶ Сборка генома (строковые алгоритмы, алгоритмы на графах, частотный анализ)
 - ▶ Разработка лекарств (вычислительная геометрия)
- ▶ Производство и логистика
 - ▶ Планирование логистических операций (кратчайшие пути, задача коммивояжера)
 - ▶ Оптимизация производственных операций (построение расписаний)



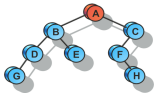
Зачем нужны алгоритмы и структуры данных?

- ▶ Интернет
 - ▶ маршрутизация пакетов (поиск кратчайших путей)
 - ▶ фильтрация спама (фильтры доменов)
 - ▶ функционирование веб-серверов (базы данных, строковые алгоритмы)
- ▶ Медицина и биология
 - ▶ Сборка генома (строковые алгоритмы, алгоритмы на графах, частотный анализ)
 - ▶ Разработка лекарств (вычислительная геометрия)
- ▶ Производство и логистика
 - ▶ Планирование логистических операций (кратчайшие пути, задача коммивояжера)
 - ▶ Оптимизация производственных операций (построение расписаний)
- ▶ Электронная коммерция и финансы
 - ▶ Защита информации (алгоритмы криптографии)
 - ▶ Оперативное реагирование на события рынка (машинное обучение)



Алгоритмы и структуры данных в быту

Пример — корзины для сортировки одежды

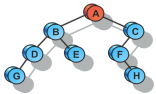


Алгоритмы и структуры данных в быту

Пример — корзины для сортировки одежды



<http://www.cleanlaundry.com/sorting-secrets-revealed/>



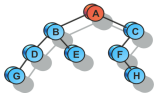
Алгоритмы и структуры данных в быту

Пример — корзины для сортировки одежды



<http://www.cleanlaundry.com/sorting-secrets-revealed/>

- ▶ «Данные» — одежда того или иного цвета



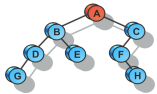
Алгоритмы и структуры данных в быту

Пример — корзины для сортировки одежды



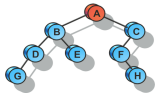
<http://www.cleanlaundry.com/sorting-secrets-revealed/>

- ▶ «Данные» — одежда того или иного цвета
- ▶ Операции:
 - ▶ **оценка** количества одежды заданного цвета
 - ▶ **добавление** произвольного предмета одежды
 - ▶ **извлечение** всей одежды заданного цвета



Закключение

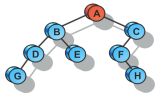
На этой лекции



Заключение

На этой лекции

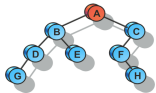
- ▶ Что такое алгоритм и структура данных



Заключение

На этой лекции

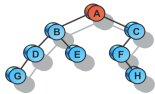
- ▶ Что такое алгоритм и структура данных
- ▶ Примеры алгоритмов и структур данных



Закключение

На этой лекции

- ▶ Что такое алгоритм и структура данных
- ▶ Примеры алгоритмов и структур данных
- ▶ Зачем нужны эффективные алгоритмы и структуры данных?
Алгоритмы как технология



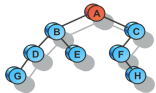
Заключение

На этой лекции

- ▶ Что такое алгоритм и структура данных
- ▶ Примеры алгоритмов и структур данных
- ▶ Зачем нужны эффективные алгоритмы и структуры данных?
Алгоритмы как технология

На следующей лекции

- ▶ Как измерять эффективность алгоритмов? O-нотация



Заключение

На этой лекции

- ▶ Что такое алгоритм и структура данных
- ▶ Примеры алгоритмов и структур данных
- ▶ Зачем нужны эффективные алгоритмы и структуры данных?
Алгоритмы как технология

На следующей лекции

- ▶ Как измерять эффективность алгоритмов? O-нотация

Спасибо за внимание!