C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

C++:

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С

STL:

* Виды алгоритмов
* Виды контейнеров
* Виды итераторов
* Функторы

ООП:

* Классы и структуры
* Конструкторы, деструкторы
* Члены и друзья классов, спецификаторы доступа
* Наследование
* Виртуальные функции
* SOLID
* Шаблоны проектирования

Алгоритмы, структуры данных:

* Сортировки
* Поиск
* Деревья – (сбалансированные) деревья бинарного поиска, черно-красные деревья, АВЛ, двоичные
* Графы

Git:

* Создать/клонировать репозиторий origin
* Сделать add, rm, commit, reset, stash
* Взаимодействие с удаленным репозиторием pull, push, fetch
* Просмотреть статус/разницу текущего состояния проекта
* Создать новую ветку
* Слить ветки

<http://gitref.org/basic>

<https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

<http://git-scm.com/book/en/v1/Git-Branching-What-a-Branch-Is>

<https://sandofsky.com/images/fast_forward.pdf>

Shell, make:

* Shell crash course
* Make
* VS, Git
* Перейти на работу в linux

ОС:

* Процессы (fork)
* Прерывания, системные вызовы

Идеи «показательного» кода:

* Написать свое РВ
* Написать демон, сортирующий entries директории по частоте посещения

<http://shahmirj.com/blog/beginners-guide-to-creating-a-daemon-in-linux>

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

* Написать программу, которая сортирует содержимое любого файла или списка файлов за максимально короткое время с выводом параметров (время, доп.место, число сравнений, обменов) использованной и альтернативных видов сортировок
  + Вычищать файл от незначащих символов <http://stackoverflow.com/questions/1894886/parsing-a-comma-delimited-stdstring>
  + частотный анализ текста <http://habrahabr.ru/sandbox/90279/>
  + Написать программу автоматического распознавания типа данных элементов сортируемого файла

**Задание:**

Загрузка и преобразование файла в вектор

Подставить функции сортировки

<https://hbfs.wordpress.com/2011/03/01/shellsort/>

Печать результатов, в т.ч.:

* Замеры времени
* Сбор статистики – cmp, moves
* Формат вывода

Параметризовать тип данных последовательности

Выгрузка отсортированной последовательности в файл

Потестить в Ubuntu, сделать в виде утилиты (с ключами-параметрами), сделать make, запаковать, сделать sh

Написать компаратор сравнения строк

Анализ, какой алгоритм лучше в случае разных данных

Анализ типа символов (что если смешанный?)

Расположение членов класса, разбивка на функции

Исключения

Комментарии

README

Изучить свойства утилит sort и понять, как можно дополнить.

<http://git.savannah.gnu.org/cgit/coreutils.git/tree/src/sort.c>

Сделать код переносимым (в POSIX системах и др.)

* Типы данных
* Пространства имен
* Явные преобразования типов
* Функции, перегрузка функций
* Перегрузка операторов
* Исключения
* Динамически размещаемые объекты
* Работа с потоками ввода/вывода

<http://stackoverflow.com/questions/9040651/c-file-handling-is-open-returning-bad> - загрузка из файла

<http://www.cplusplus.com/forum/general/3221/> - как делать переносимый код для считывания нужной директории посредством ОС

* Отличия С++ от С