Практика, архиватор Хаффмана

Юрий Литвинов
y.litvinov@spbu.ru

03.11.2023

Замечания по домашним работам (1)

- Внутренние функции модуля — static
  - Это делает функции невидимыми для линковщика вне модуля
  - Бывают ещё static-переменные — не очищаются при выходе из функции
  - См. https://en.cppreference.com/w/c/language/storage_duration для тайных знаний
- Явно приводите указатель к нужному типу после malloc/calloc и т.п.
- Не выделяйте большие массивы на стеке!
- Сдавайте задачи только пуллреквестами
- Всё, что по смыслу размер или индекс — size_t
- Подключайте нужные хедеры
  - Например, string.h для strcmp

Замечания по домашним работам (2)

- Константные указатели: **const** Type * **const**
- var++ vs ++var
- Используйте модули везде
  - Тесты в отдельном модуле
  - Структуры данных в отдельном модуле
  - Файловый ввод-вывод (и сложный UI) в отдельном модуле
  - Бизнес-логика (код, который собственно решает задачу) — в отдельном модуле
  - И main
- Читабельные имена веток/PR на английском, никакого транслита
- Не коммитьте лишние файлы!
- Не дублируйте код
- И главное — не сдавайте новые задачи, не поправив старые и повторяя старые ошибки

Замечания по домашним работам (3)

- Замер времени:

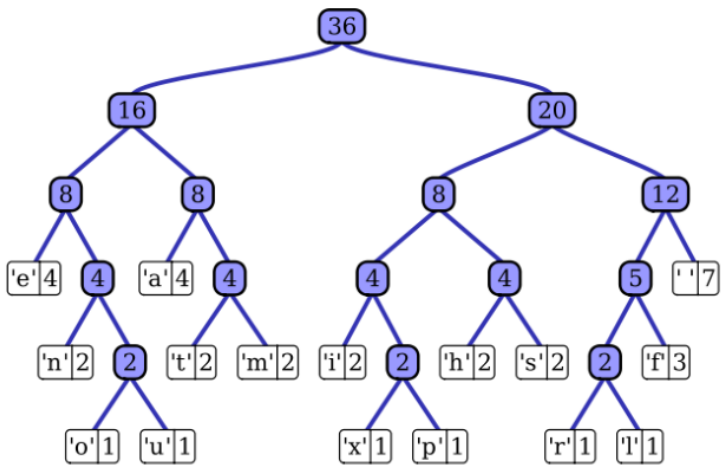
```
typedef int *(SortMethod)(size_t const * const, size_t);

double calcDuration(SortMethod method, const * const array,
    const size_t arraySize)
{
    ...
}
```

- Примеры: https://learnc.info/c/function_pointers.html
- Чтение строк произвольной длины:
  - https://programforyou.ru/poleznoe/effektivnoe-poluchenie-stroki-proizvolnoj-dliny-na-c-i-c-plus-plus
- Можно сделать string {**char** *data, **size_t** size, **size_t** capacity} и читать в неё (автоматически реаллоцируя по необходимости)

Напоминание идеи алгоритма

- Считаем частоты символов во входной строке
- Склеиваем два самых редких в один псевдосимвол, пока не получили одно дерево
- Строим по дереву префиксный код в виде таблицы
- Бежим по строке, заменяя символ на его код
  - Используем битовый буфер для формирования байта выдачи
- При разархивировании бежим от корня до листа



Пример: “this is an example of a huffman tree”

Задача

1. Описать двоичное дерево, хранящее символы и частоты, в отдельном модуле
2. Построить дерево частот
3. Построить таблицу с префиксными кодами
4. (*) Получить сжатую строку
5. (*) Оценить степень сжатия
6. (*) Реализовать разжатие