Проекты

Структуры данных

Существует большое множество различных структур данных. У каждых свои преимущества и недостатки. Тем не менее, они играют важнейшие роли в организации памяти самых различных программ.

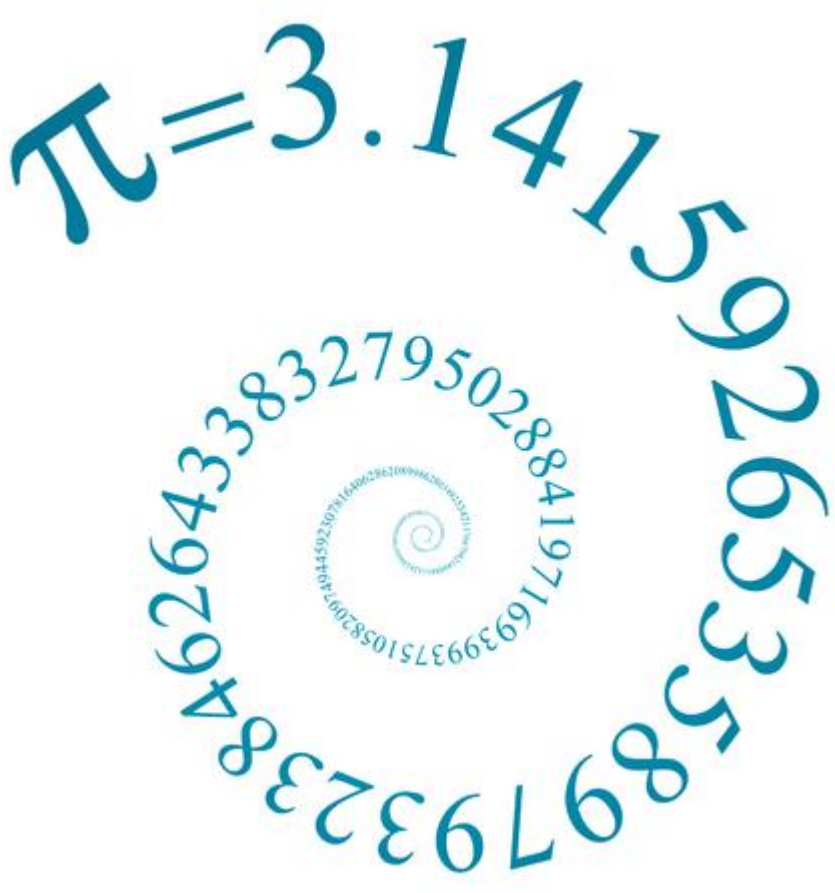
Предлагается реализовать одну из структур данных:

- ~~Массив~~
- ~~Динамический массив~~
- ~~Списки~~
- ~~Двоичное дерево поиска~~
- ~~Стек~~
- ~~Очередь~~

Любую, кроме этих

Структура данных	Временная сложность								Сложность по памяти
	В среднем				В худшем				В худшем
	Индексация	Поиск	Вставка	Удаление	Индексация	Поиск	Вставка	Удаление	
Обычный массив	O(1)	O(n)	-	-	O(1)	O(n)	-	-	O(n)
Динамический массив	O(1)	O(n)	O(n)	O(n)	O(1)	O(n)	O(n)	O(n)	O(n)
Односвязный список	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Двусвязный список	O(n)	O(n)	O(1)	O(1)	O(n)	O(n)	O(1)	O(1)	O(n)
Список с пропусками	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n log(n))
Хеш таблица	-	O(1)	O(1)	O(1)	-	O(n)	O(n)	O(n)	O(n)
Бинарное дерево поиска	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)	O(n)	O(n)	O(n)	O(n)
Декартово дерево	-	O(log(n))	O(log(n))	O(log(n))	-	O(n)	O(n)	O(n)	O(n)
Б-дерево	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Красно-черное дерево	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)
Расширяющееся дерево	-	O(log(n))	O(log(n))	O(log(n))	-	O(log(n))	O(log(n))	O(log(n))	O(n)
АВЛ-дерево	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(log(n))	O(n)

Вычисление числа Пи



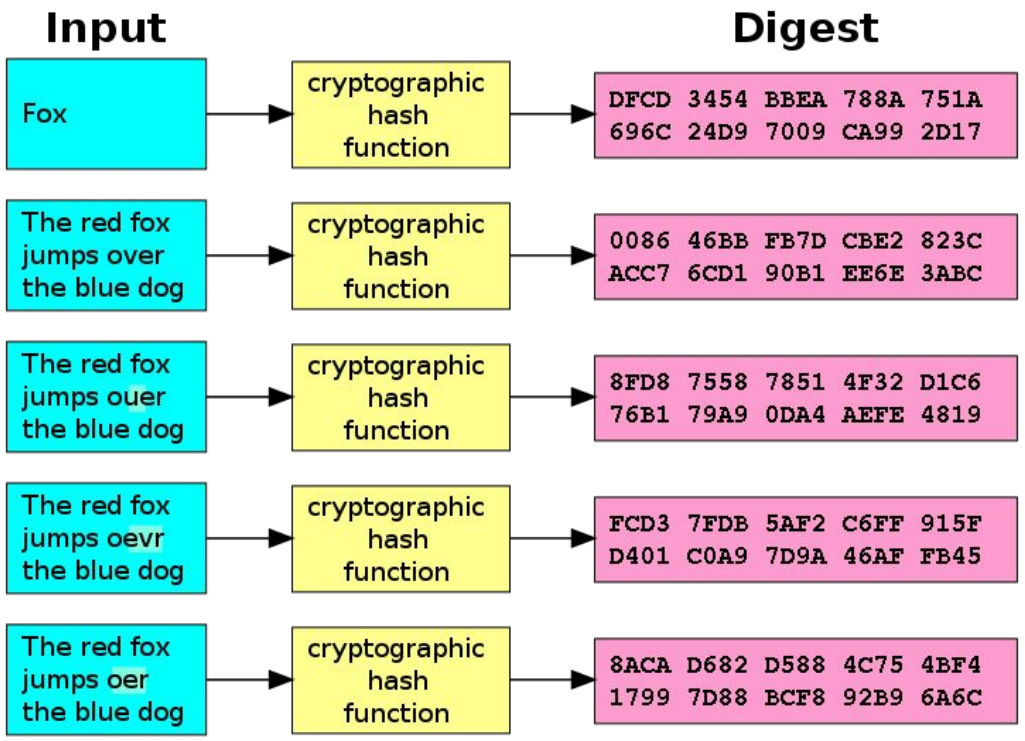
Число пи по своей природе является бесконечным. Это дает некоторый интерес для его изучения. Существует множество различных алгоритмов нахождения цифр этого числа. Предлагается найти число Пи с заданной точностью для каких-либо дополнительных целей. (например, поиска некоторой последовательности)

Алгоритм шифрования

Реализуйте один из актуальных алгоритмов шифрования.

- AES
- ГОСТ
- RSA
- …

Хеш-функция



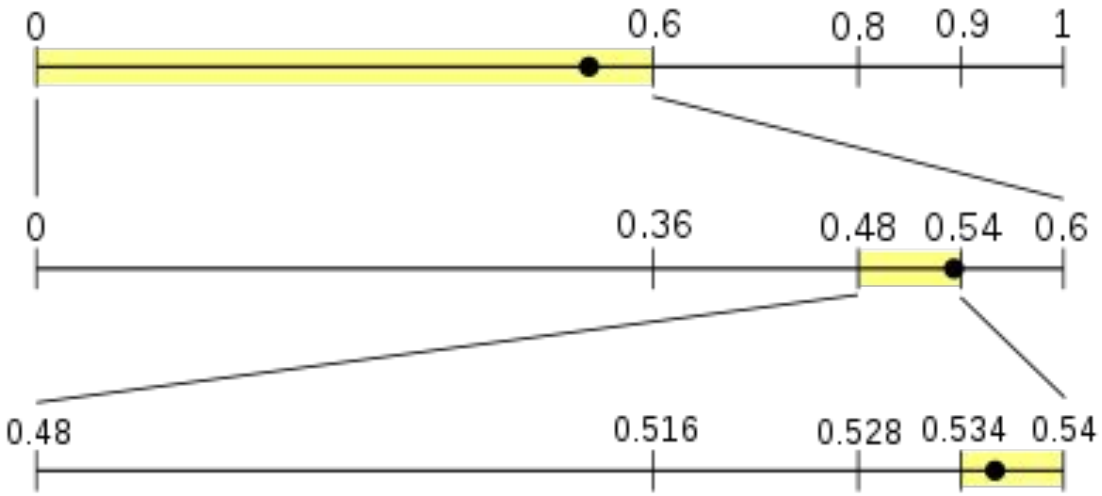
Пусть имеется некоторый блок данных, для которого требуется найти характеризующее его значение некоторого конкретного размера. Функция, которая возвращает такое значение называется хеш-функцией. Кроме того, есть криптографические хэш-функции, результат которых крайне-крайне-редко повторяется. Область применения таковых может быть различная: идентификаторы файлов, способ надежного хранения паролей, цифровая подпись, контрольная хеш-сумма и т.д. Задача состоит в том, чтобы реализовать одну из таких актуальных криптографических хеш-функций.

Генератор цепей Маркова

Возьмите некоторый текст и запишите, какие слова встречаются вслед за какими. Получится, некоторое дерево, где если идти от стартовых слов к завершающим, получаются предложения, которые принято называть псевдо осмысленными. Предлагается написать генератор цепей Маркова (минимум 2-го порядка)

Арифметическое кодирование

Один из самых оптимальных методов сжатия информации. Позволяет представлять данные в виде вещественного числа от 0 до 1.

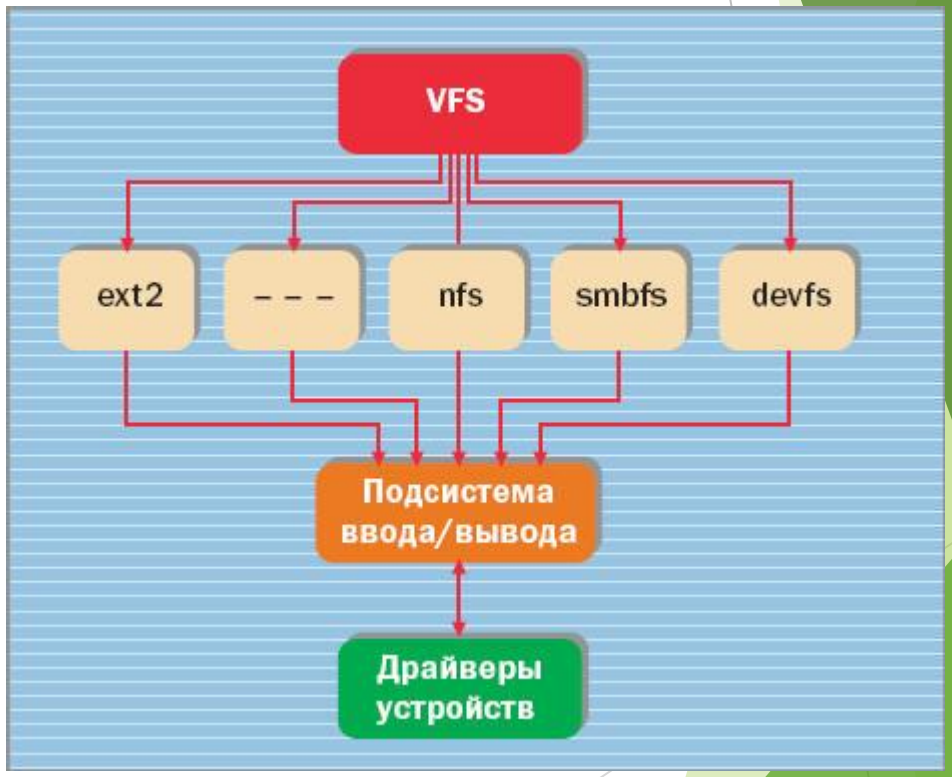


Блокчейн (криптовалюта)

Реализуйте свою локальную «криптовалюту» на технологии блокчейн (blockchain).

Виртуальная файловая система

Реализуйте консольную оболочку, в которой будет возможна работа с виртуальными файлами, с возможностью сохранения в виде бинарного файла в основной ОС.



База данных

Не много, не мало – реализуйте свою консольную программу для работы с локальной базой данных.

Эмулятор процессора

Реализуйте программу, которая эмулирует работу некоторого процессора, способного выполнять вычисления и взаимодействовать с пользователем.