

Разложить число на простые делители

## Декодирование АОН

- Нужно преобразовать строку по следующим правилам:
- Если символ идет 2 и больше раз подряд - записать его в результат 1 раз
- Если символ повторяется 1 раз - отбросить
- Если повторяется два и более раз - последний символ, записанный в результат записать еще раз
- "" => ""
- "1" => ""
- "11" => "1"
- "11111" => "1"
- "11" => "1"
- "11" => "11"
- "11122234###55" => "1225"

Гномья сортировка Сравниваются соседние элементы. Если они неупорядоченны - они меняются местами и делается шаг назад. Если они упорядоченны, то шаг вперед. Если дошли до конца, то сортировка окончена.

Двоичный поиск Найти элемент в упорядоченном массиве методом дихотомии.

## Кодирование Шеннона — Фано

- Метод выбора близкого к оптимальному кода для сообщения
- Сообщение бьется на элементы
- Изначально коды для всех элементов пустые
- Элементы множества выписывают в порядке убывания вероятностей.
- Множество делится на две части, суммарные вероятности символов которых максимально близки друг другу.
- К коду первой половины элементов дописывается "0", второй "1"
- Алгоритм повторяется для обеих частей
- \* Искать оптимальное разбиение

9) кодирование и декодирование файла по Хаффману. На диске есть файл с именем "input.txt". Его нужно прочитать, закодировать символы используя алгоритм Хаффмана и записать результат в output.bin.

В решении должно быть две функции  $hf_{encode}(string)str \rightarrow str$ ,  $hf_{decode}(string)str \rightarrow str$ , — ..

Интерпретатор minilisp Программа на mini-lisp имеет вид (oper param1 param2 para3 .... paramn), здесь oper - имя функции - любой набор символов, кроме пробелов; param2 - целое, строка в кавычках (без кавычек внутри) или другая программа на mini-lisp. Допустимые oper - + (складывает все операнды), - (вычитает из первого все операнды), print (печатает все операнды через пробел). Нужно исполнить программу, переданную на вход.

- (+ 1 2 3) => 6
- (print (+ "a" "bc")) => печатает abc

интерпретатор подмножества языка forth

Программа на Forth состоит из набора команд(слов), некоторые из которых имеют параметры. Для хранения данных используется стек - команды получают свои операнды с вершины стека и туда же сохраняют результаты. В подмножестве 5 команд:

- put значение - ложит значение на вершину стека. Значение может быть числом или строкой. Строка заключается в кавычки, внутри строки кавычек быть не может
- pop - убирает значение с вершины стека
- add - изымает из стека 2 значения, складывает их и заносит результат в стек
- sub - изымает из стека 2 значения, вычитает их и заносит результат в стек
- print - вынимает из стека 1 значение и печатает его.

Каждая команда начинается с новой строки. Строки начинающиеся с '#' - комментарии. Программа должна содержать функцию `evalforth()`, `forth.maineval`  
Например если в `example.rft` будет:

```
put 1  
put 3  
25  
add  
print
```

Программа должна напечатать '4'. Сложение имеет такой же смысл, как и в питоне. Вычитание для строк не определено



строковые функции Написать строковые функции `xfind`, `xreplace`, `xsplrit`, `xjoin` используя срезы строк (без применения других методов строк).

- `xfind(s1, s2) == s1.find(s2)`
- `xreplace(s1, s2, s3) == s1.replace(s2, s3)`
- `xsplrit(s1, s2) == s1.split(s2)`
- `xjoin(s, array) == s.join(array)`

Умножение больших чисел Реализовать алгоритм Карацубы для умножения больших чисел.  $AB * CD == (A + B * 2^m) * (C + D * 2^m) = A * C + 2^{2*m} * B * D + 2^m * (B * C + A * D) = A * C + 2^{2*m} * B * D + 2^m * ((A + B)(C + D) - A * C - B * D)$

## Алгоритм Кнута-Морриса-Пратта

-

Тетрис

Острова Задан двумерный массив из 0,1. Островом называется связанная группа единиц, в т.ч. по диагонали. Посчитать количество островов.