



Лекция №1

Сложность, тестирование,
особые случаи

О чём курс?

- научиться писать эффективный код
- умение тестировать
- . . .

Hints:

- 10.000 строк эффективного кода

2 лекции. После лекции контест - 10 задач

Сложность — порядок кол-ва действий

$\mathcal{O}(N)$

Э пространственная сложность — кол-во памяти

Задача

Дана строка UTF-8

вывести самый частый символ

Решение

Тривиальное: цикл в цикле с счётчиками $\sim \mathcal{O}(N^2)$

Оптимальное: $\mathcal{O}(N)$

K
кол-во уникальных букв (создание за $\mathcal{O}(N)$)

1) заводим сет уникальных букв $\mathcal{O}(N)$

2) пробегаем строку и заполняем массив ($\Omega(N)$), скажем глобальный счетчик

Особые случаи

- если убрать $ans = ''$ (каких удалить проверка не нужна - ответ)
- сразу проверять $len == 0$ нет смысла (логика обрабатывает)

Тестирование

- условие !!!
 - обычный случай
 - подводные камни
- пустой ввод 1 символ

Вырожденная последовательность

Пример

Задача

Даны a, b, c — целые

$$ax^2 + bx + c = 0$$

Вывести корни в порядке возрастания

Решение

- Типичные ошибки: Приоритет операций

$[d \leq 0]$ не обработан

$[a=0]$ не обработан

$[b=0]$ не обработан

[∞] число решений

[порядок вывода]

[помогающая отладка]

Лекция №2

Линейный поиск

Классические задачи

Задача №1

Дана последовательность чисел длиной N

Найти левое вхождение $X > 0$
или -1 .

Решение

① default answer = -1

② if $ans == -1$ and $cur == X$:
 $ans = i$

* желательно оформлять в виде func,
а ввод-вывод отдельно

Задача #3

Из последовательности найти максимум.

Решение

- ⑤ В общем слугает
Стандартный прием: минимум - это 0.
- ⑥ На каждом шаге сравниваем с
ответом.
- ⑦ Уменьшаем количества:
вместо сравнений по индексам
сравниваем по индексам.
(но страдает время)

Задача #4

Найти max и предыдущееимальное
(Базн: храним, потом взаимный обмен)

Задача #5

Найти четкий минимум
или -1

Задача #6

Вызвести 2 самых коротких слова
из по след-ти

Решение

- ① → исход минимум

② → ищем 2-ую строку такой же длины

Python

- $+=$ конката строка, новою
- через список, через `"join(ans)` получим ответ.

Задача #7

Решение

Есть остров 2D. Определить число
смешанных квадратов.

① Ищем макс. высоту

② Храним разность с последующими
столбиками



③ Универсалько даже для > максимумов.

Задача с соседством

Дана строка из букв A-Z

Сжатие RLE

$AAA \rightarrow A3$

Решение

- ① Стока из уникальных символов
- ② lastpos и lastsym для ответов.

Лекция №3

Множества

Как устроено мк-во?

- добавить
- удалить
- проверить наличие

Кусок лекции 8
Структура
данных

Пример функиций:

$F: X \rightarrow X \% 10$, то есть коллизии: 1%.

Решение: список.

В python:

1) список списков

2) поиск (линейное время)

добавление (append)

удаление (swap + pop)

Что хранить эффективно?

- Только неизменяемый объект
- Хэши-функция с равномерным распред.

Амортизированная сложность.

- ↑ размер $\Rightarrow \uparrow$ память $\underline{\mathcal{O}}(N)$
- ↓ размер $\Rightarrow \uparrow$ время

Решение: капотило $\Rightarrow \text{size} *= 2$.

Примеры задач

Задача 5 / Дано последовательности N ,
дан X . Найти 2 различных
числа с суммой X

Решение:

$\mathcal{O}(N)$:

- при добавлении a в set
проверим наличие $X-a$.

Задача 2 \checkmark Так словарь N слов, длина каждого $\leq k$

Проверить все исходные слова с возможными пропуском 1 буквы в словаре.

Решение

$\mathcal{O}(NK + M)$:

k строк за k^2 , т.к.

создание строки

за $k \Rightarrow \mathcal{O}(Nk^2 + M)$

- в словарь кладем все возможные слова с ≤ 1 пропуском.
- проверяем каждое.

Лекция № 4

Словари

и
сортировка подсчетом.

- отсортировать range($k+1$)

Тогда Э сортировка за $\mathcal{O}(k)$, а не

$\mathcal{O}(k \log k)$

```
tmp = np.zeros( $\ell$ )
for i in range(len(mas)):
    tmp[mas[i]] += 1.
```

Задача №1

Даны X и Y без ведущих 0
Можно ли получить первое из
второго?

Решение:

Сортировка подсчета + сравнение
массивов.

Задача №2

Доска шахматная $N \times N$,

M ладей

Сколько пар ладей будут
грыз грыза?

Решение:

- Словарь с строкой/столбцом:
сколько ладей в полоске
- $\sum (val - 1)$ для словарей - ответ.

Задача №3

Дана строка S

Вывести частоты
частоты букв.

Решение

- считаем частоты σ
- находим самую частоту и делим от неё до 1
- по всем буквам: если $\sigma >$ счетчика, то #, иначе пробел.

Критерии кач-ва алгоритмов:

- память
- время на реализацию
- сложн-ть поддержки
- возм-ть параллельного вычисл
- квалиф-ия сотрудника
- стоимость оборудования

Задача №4

Группировка слов по одинаковым буквам

Решение

- сортировка букв + словарь

Ключ - само слово

! Сортировка за $N \log N$

через сортировку подсчетом, но

↑ "Тормознутость"

Лекция № 5

| Префиксные суммы

Дан массив и посчитать промежуточные суммы.

nums	5	3	8	1
sum	0	5	8	16

$$\text{sum}[i] = \text{num}[i-1] + \text{sum}[i-1]$$

$$\text{sum}[2, 5] = \text{sum}[5] - \text{sum}[2]$$

Задача № 1

запросы: сколько купе в интервале

Решение

префиксское кол-во купеи.

Задача №2

запрос: сколько интервалов складской суммой?

Решение:

ответ: когда префиксные суммы равны.

словарь с ключами — префиксными суммами
посчитали, добавили 1 к счетчику

два указателя

Задача №3

дана отсортированная посл-ть и k.

кол-во пар $A, B: B - A > k$.

Решение:

- ① фиксируем левый указатель
- ② двигаем правый, до каких-то
с позиции, на которой остановился
правильный

Задача №4

1) 2 самых слабых в сумме, ке хуже
лучших
для отсортированного массива
найти максимум при пресекомализи
 $(\sum \text{ так, чтобы выполнялось } i)$

Решение

- ① Выбираем слабейшего
- ② Характеристика самых слабых
- ③ Дераем всех между ними
- ④ Самый слабый:
дераем элем-т и след-ий.
Пока $\sum >$, тодвигаем право.

Задача №5

Все отсорт-ые пост-ти слив в одну

Решение:

Как в merge sort