Набор монеток — см. сайт, репозиторий с тестами.

возр. слож-ть ⟶

1. in ⟶ 1.xyz.out ← 2 в-та
   ввод   1.abc.out   задачи,

проще и
сложнее.

$O(n^2)$ ← не пройдёт ~1000000 чисел

$O(n \log n)$ ← ~30·000000

- алг. со строками (edit distance)
- выч. геометрия.
- big data: ✓ mmds — google

Со строками.
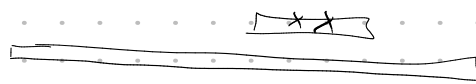
1) подстр. в строке
2) edit distance

Поиск подстроки в строке.

Дано s ← что искать

t ← где искать

⟶ ответ s есть с 3-ей позиции

$s = abc$

$t = xyzabcxyz$
    0 1 2 3 4 5 - - -

Наивно t

0)
1)
2)
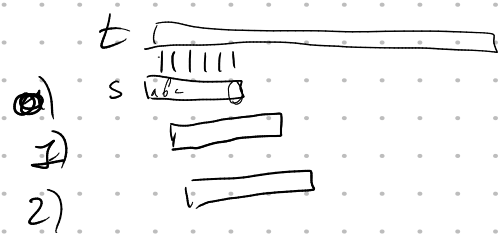
сложность в худшем случае

$$O(s(t-s)) = O(st)$$

если $t >> s$

Алгоритм Кнута-морриса-пратта (КМП)

ф-ия $p(s)$ от строки, длина max префикса =
                                    постфикса.

$p(\overline{abc}\ xyz\overline{abc}) = 3$

$$p(\underline{a}\,xyzba\,cx\underline{a}) = 1$$

$$p(\underline{abc}) = 0$$

0...k-1

$$p(\overset{k}{\underbrace{\square}}\overset{k}{]}\;\overset{k}{\square}\;x) = \begin{cases} S[k]=x \text{, то} \quad p(s)+1 \\ \qquad\qquad = k+1 \\ \\ \cdots \end{cases}$$

как изменится значение ф-ии, если к строке добавить символ?

$$p(\overset{k}{\underbrace{\square}}\;\overset{k}{\square}) = k$$
$$\quad s$$

$$p(\boxed{\underset{\ell}{\text{▨}\,|x}}\;\cdots\;[\overset{-k-}{\underset{\ell}{\text{▨}|x}}) \quad -$$

▨ — это префикс s = постфикс S

— это префикс k = постфикс k.

$$p(sx) = \ell$$

вычисляем $p(sx)$

$k = p(s)$. Если $S[k]=x$, то ответ $k+1$

иначе

$K := p(S[0..k-1])$ Если $S[k]=x$, то ответ $k+1$

иначе

$K := p(S[0..k-1])$

если ничего не подошло, ответ $:0$

Поиск подстроки S в строке t:

$\underline{d = S\#t}$ — создаём строку, где # — новый символ не из s и не из t.

Вычисляем постепенно $p(d[0])\quad p(d[0..1])$

$\cdots\; p(d[0..\text{конца}] = S\#t)$.

Когда $p = |s|$, нашли совпадение.

Анализ: $O(|s|+|t|)$ — сложность.
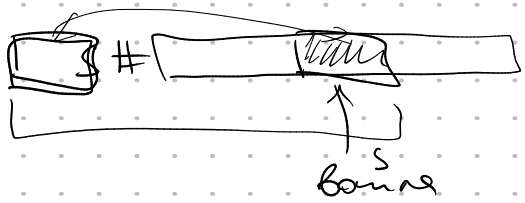
Тест: *.in   2 строки, чтение s, потом t
                символы a-z

                abc
                xyzabcxyzabxyz

        *.out
                3   — индекс первого вхождения.

____

Edit distance
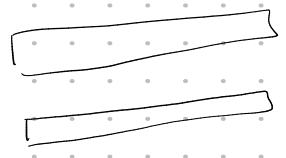
s
t

похожесть в виде числа.

— исправление опечаток
— неточный поиск
— плагиат

Операции: заменить букву x на y
          вставить букву, удалить букву

$abcdef \xrightarrow{x c} abdef \xrightarrow{f \to g} abdeg$

$abdeg \xrightarrow{\sim d} abddef \xrightarrow{d \to e} abdeef \rightarrow$

$\xrightarrow{x e} abdef \xrightarrow{f \to g} abdeg$

$D(s,t) = \min$ кол-во действий для превращения s в t.

$D(abcdef, abdeg) = 2.$

Утв. D — это расстояние $D(s,s)=0$ $D(s,t) \geq 0$

$D(s,t)=0 \Rightarrow s=t$

$$D(s,t) + D(t,u) \geq D(s,u) \qquad \text{—очевидно}$$

$$\underset{\alpha}{} \qquad \underset{\beta}{}$$

$$s \xrightarrow{\alpha} t \xrightarrow{\beta} u$$
$$\underbrace{}_{\alpha + \beta}$$

Вычисление.
$$s = s_0 s_1 \dots s_{m-1} \qquad |s| = m$$
$$t = t_0 t_1 \dots t_{n-1} \qquad |t| = n$$

Дин. фия:
$$d(i,j) = D(s_0 s_1 \dots s_{i-1}, \; t_0 t_1 \dots t_{j-1})$$

сразу:
$$d(i,0) = i \quad \Leftarrow i \quad \text{удалений}$$
$$d(0,j) = j \quad \Leftarrow j \quad \text{добавлений}$$

$$\boxed{d(i+1, \; j+1)}$$

← посл. буква.
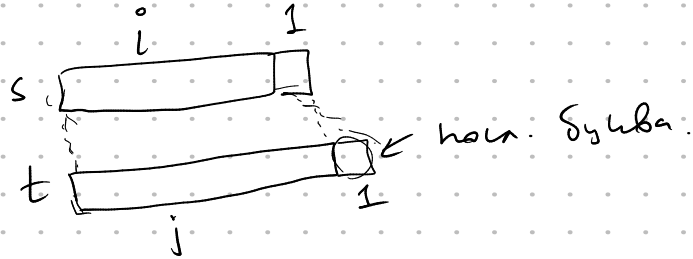
как получить
букву t[j]

$$= \min \begin{cases} d(i,j) + \begin{matrix} 0 \text{ при } s_i = t_j \\ 1 \text{ при } s_i \neq t_j \end{matrix} \quad \Leftarrow \text{ замена} \\[2mm] d(i+1, j) + 1 \quad \Leftarrow \text{ добавление буквы } t_j \\[2mm] d(i, j+1) + 1 \quad \Leftarrow \text{ удаление буквы } s_i \end{cases}$$

пример.

$s_i \quad abcde$

$t_j \quad abdeg$

i \ j	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	0	1			
2						
3						
4						
5						

← $d(0,j) = j$

$a \quad d(i,0)=i$

$d(1,0) \quad a$

$d(i,j) \longrightarrow$

$$d(a,a) = \min \begin{cases} d(-,-) + 0 = 0 \\ d(a,-) + 1 = 1+1 = 2 \\ d(-,a) + 1 = 1+1 = 2 \end{cases} = 0$$

$$d(1,2) = d(\underset{0}{a}, \underset{}{ab}) = \min \begin{cases} d(-, a) + 1 & a \neq b = 2 \\ d(a, a) + 1 & = 1 \\ d(-, ab) + 1 & = 2+1 = 3 \end{cases} = \underline{1}$$

Получаете, что $d(i, j+1)$ вычисляете через

$\underline{d(i,j)} \quad d(i+1, j)$ и $d(i, j+1)$.

Проще всего хранить двумерный массив

d:
     for $i = 0$ to m
       for $j = 0$ to n
$\longrightarrow$ $d[i,j] = \dots$ через $d[i-1, j] \quad d[i-1, i-1]$
                      $d[i, j-1]$.
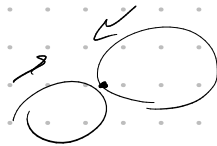
по строчное заполнение массива.
потом по массиву можно восстановить
действия.
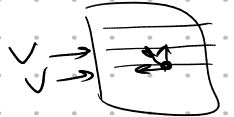
Если восстановление не нужно, можно
хранить только 2 последние строки массива.

Сложность $O(m \times n)$.

<u>Выч. геометрия.</u>

-графика   -физика   -карты
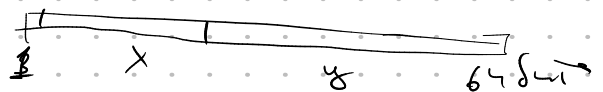
Точная вычислений: вещ числе хранятся неточно.

0.1 + 0.2
0.3 00000000...04
$= X \cdot 2^y$ $0.1 \neq \frac{X}{2^y}$

8байт -число

$\underbrace{1}_{} \quad \underbrace{\quad X \quad}_{} \quad \underbrace{\quad \quad \quad y \quad \quad \quad}_{64бит}$

В геометрии, если числа вещественные
$\varepsilon = 1e-10 \quad \quad (1e-8)$

if $a == b$ $\iff$ if $|a-b| < \varepsilon$

if $a > b$ $\iff$ if $a > b + \varepsilon$ $\checkmark$

if $a \geqslant b$ $\iff$ if $a \geqslant b - \varepsilon$

$$a \geqslant b, \quad b \geqslant a$$

$0.1 + 0.2 == 0.3$ $\to$ False

$$\overset{\cdot\;\;\overline{\,-\,}}{a \;\; a+\varepsilon \;\; a+2\varepsilon}$$
$$a = a + \tfrac{1}{2}\varepsilon$$
$$a + \varepsilon = a + \varepsilon \qquad \text{но } a \neq a + \varepsilon$$

2). Все моорg sensie, тогда можно избежать потери

точности:

$$\frac{a}{b} == c \iff a == c \cdot b$$

$$\sqrt{a} > \sqrt{b} \iff a > b$$

$$\sqrt{a} > b - 1 \iff \; \approx \; a > (b-1)^2$$

но аккуратно с переполнением.
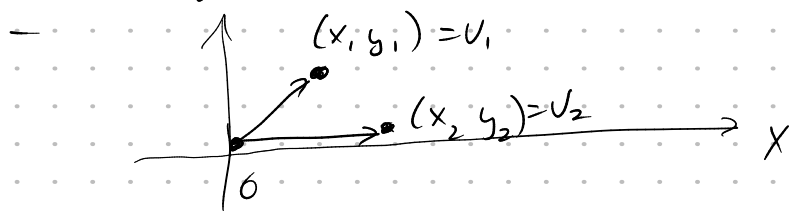
$$V_1 \cdot V_2 = x_1 x_2 + y_1 y_2$$
$$= |V_1| \cdot |V_2| \cdot \underline{\underline{\cos \alpha}}$$
$$|V_1| = \sqrt{x_1^2 + y_1^2}$$

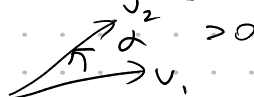$$\cos \alpha = \frac{V_1 V_2}{|V_1||V_2|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{\cdots}\;\sqrt{\cdots}}$$

$$\begin{vmatrix} x & y & z \\ x_1 & y_1 & 0 \\ x_2 & y_2 & 0 \end{vmatrix} = z(x_1 y_2 - x_2 y_1)$$

$$V_1 \times V_2 = x_1 y_2 - x_2 y_1$$
$$\overset{\shortparallel}{\;} |V_1||V_2| \sin \alpha$$

$$\sin \alpha = \frac{x_1 y_2 - x_2 y_1}{|V_1||V_2|}$$

Замечание

$\alpha > 0$

$$V_1 \times V_2 = -V_2 V_1$$

$\alpha < 0$ $\quad V_2 V_1$

$$\sin \alpha = -\sin -\alpha$$

Позволяете возможность определять слева/справа:

$(x_2, y_2) = v_2$

$(x, y_1) = v_1$

0

$v_1 \times v_2 > 0$ — слева

$v_1 \times v_2 < 0$ — справа.

Еще мыслы.

$(x_1, y_1) = A$

$(x_2, y_2) = B$
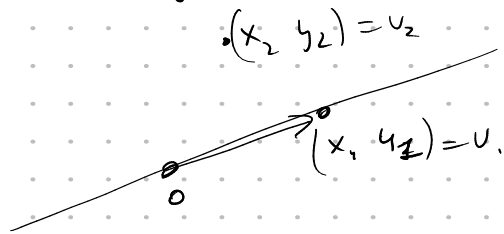
вектор $\overline{AB}$ имеет координаты $B - A = (x_2 - x_1, y_2 - y_1)$

Задача Вычислить площадь многоугольника.

$P_2 : x_2, y_2$

$P_1 = x_1, y_1$

x_3, y_3

x_4, y_4

x_5, y_5

$(0,0)$

x_6, y_6

$S = ?$

$S_1, S_2, S_3, S_4, S_5, S_6, S_7$

S_3

S_4

0

S_2

S_1
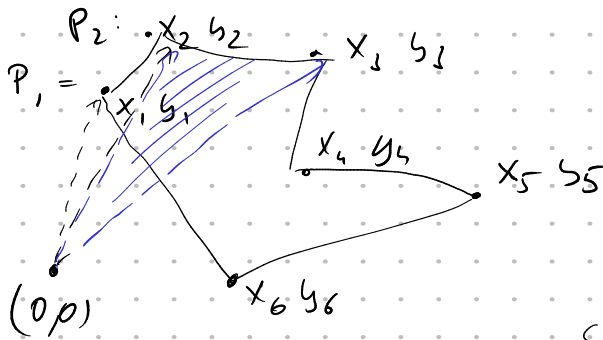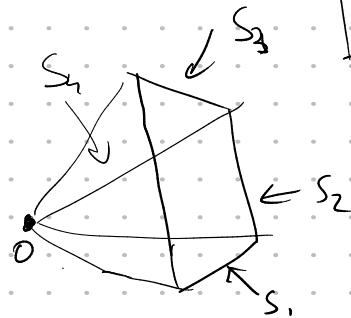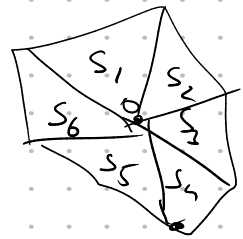
$\frac{1}{2} P_1 \times P_2 = \underbrace{\frac{1}{2} |P_1| \cdot |P_2|) \sin \alpha_{12}}_{\text{площадь } \Delta}$

$= \frac{1}{2} (x_1 y_2 - x_2 y_1)$

$\frac{1}{2} P_2 \times P_3$

Ответ: $\dfrac{1}{2} \sum\limits_{i=1}^{n} P_i \times \underbrace{P_{i+1}}_{P_{n+1} = P_1}$

Дано $(x_1, y_1), (x_2, y_2) \ldots (x_n, y_n)$

площадь n-угольника. $\frac{1}{2} (x_1 y_2 - x_2 y_1 + x_2 y_3 - x_3 y_2 +$

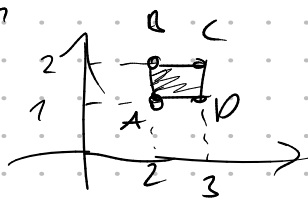$+ x_3 y_4 - x_4 y_3 + \ldots + x_n y_1 - y_n \cdot x_1)$

— в конце взять модуль.

Входные данные 1.in
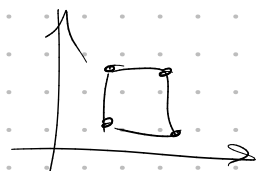
4 -точки

2 1   // A̶

2 2   // B

3 2   // C

3 1   // D.

Вывод

1

- Проверка точки внутри $\sqrt{}$ ВЫПУКЛОГО n-угольника.
  внутри
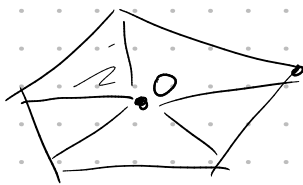
все $P_i \times P_{i+1} > 0$
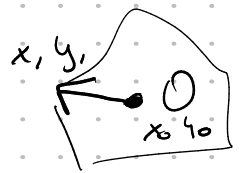
все $P_i \times P_{i+1} < 0$
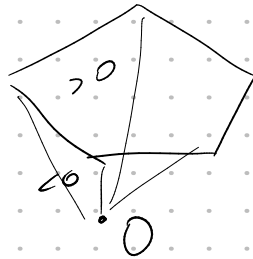
будут $P_i P_{i+1} > 0$

и $P_i P_{i+1} < 0$

1) проверил, есть ли разные знаки
   или

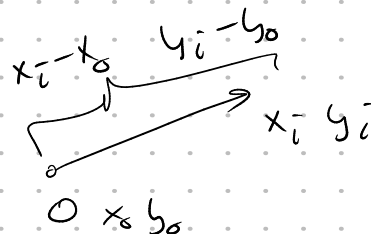2) посчитать площадь параллельно
   посчитав площадь через $\frac{1}{2} \sum\limits_i |P_i P_{i+1}|$

если точка внутри, должно совпадать.

$$S = \left| \frac{1}{2} \sum\limits_i P_i \times P_{i+1} \right| \qquad \leftarrow \text{настоящая площадь}$$

$$S = \frac{1}{2} \sum\limits_i \left| (x_i - x_0)(y_{i+1} - y_0) - (x_{i+1} - x_0)(y_i - y_0) \right|$$

Условие.

n — сколько точек

$x_1 \ y_1$

$\vdots$

$x_n \ y_n$ $\Big\}$ точки n-угольника

K

$\left\{ \begin{array}{l} x_0^{(1)} \ y_0^{(1)} \\ \\ x_0^{(k)} \ y_0^{(k)} \end{array} \right.$ — точки для проверки.

выходной файл

$\left. \begin{array}{l} IN \\ OUT \end{array} \right\}$ k ответов

IN — внутри
OUT — снаружи.

— Выпуклая оболочка.