

Алгоритмы и структуры данных-2

SET 5. Задача А3б.

Весна 2024. Клычков М. Д.

Пункт 1. Будем работать только с двухсимвольными строками. Пусть s_1, s_2 — первый и второй символы строки соответственно, $f(s_1), f(s_2)$ — ASCII коды первого и второго символа соответственно, $g(s_i) = f(s_i) - f('a') + 1$ — смещенный код i -го символа строки.

Тогда хеш-функцию можно записать как $h(s) = g(s_1) + g(s_2)p$. Требуется найти такие строки (формально вектор из двух символов), что $h(s) = 0$, то есть

$$g(s_1) + g(s_2)p = 0 \iff -g(s_1) = g(s_2)p.$$

Алгоритм поиска описанных строк при заданном параметре p будет заключаться в переборе всех возможных пар символов и проверке условия $-g(s_1) = g(s_2)p$. Заметим, что мы за при просмотре строки-пары $s = \overline{s_1 s_2}$ мы также можем рассмотреть строку $s' = \overline{s_2 s_1}$. Для понимания, как будут выглядеть значения $g(s_i)$ выпишем некоторые из них:

s_i	$f(s_i)$	$g(s_i)$
0	48	-48
1	49	-47
2	50	-46
\vdots	\vdots	\vdots
9	57	-39
A	65	-31
B	66	-30
C	67	-29
\vdots	\vdots	\vdots
Z	90	-6
a	97	1
b	98	2
c	99	3
\vdots	\vdots	\vdots
z	122	26

Теперь все готово к написанию алгоритма, воспользуемся языком программирования Python:

```
1 from string import ascii_lowercase, ascii_uppercase, digits
2
3 def find_kernel(p: int) -> list[str]:
4     symbols = list(digits) + list(ascii_uppercase) + list(ascii_lowercase) # list of symbols
5     g = lambda si: ord(si) - ord('a') + 1
6     kernel = []
7     for i in range(len(symbols) - 1):
8         for j in range(i + 1, len(symbols)):
9             if -g(symbols[i]) == g(symbols[j]) * p:
10                 kernel.append(symbols[i] + symbols[j])
11             if -g(symbols[j]) == g(symbols[i]) * p:
12                 kernel.append(symbols[j] + symbols[i])
13
14     return kernel
```

Проверим работу алгоритма:

```
1 find_kernel(13) # returns ['9c', 'Fb', 'Sa']
2 find_kernel(17) # returns ['0a']
3 find_kernel(23) # returns ['2b', 'Ia']
```

Пункт 2. Напишем код, который, пользуясь написанной функцией, выведет требуемые строки для всех значений $p \leq 31$

```
1 for p in range(-10000, 32):
2     ker = find_kernel(p)
3     if ker:
4         print(p, ker)
```

Утверждается, что порогового значения -10000 для параметра p будет достаточно, так как значения $g(s_i) \in [-48, 26]$

Вывод:

p	Values
-26	za
-25	ya
-24	xa
-23	wa
-22	va
-21	ua
-20	ta
-19	sa
-18	ra
-17	qa
-16	pa
-15	oa
-14	na
-13	ma zb
-12	la xb
-11	ka vb
-10	ja tb
-9	ia rb
-8	0Z ha pb

p	Values
	xc
-7	6Z ga nb uc
-6	0X 6Y fa lb rc xd
-5	3W 8X BZ ea jb oc td ye
-4	0T 4U 8V DY HZ da hb lc pd te xf
-3	0P 3Q 6R 9S BV EW HX KY NZ ca fb ic ld

p	Values
	oe rf ug xh
-2	0H 2I 4J 6K 8L BQ DR FS HT JU LV NW PX RY TZ ba db fc hd je lf ng ph ri tj vk xl zm
1	Fz zF Gy yG Hx xH Iw wI Jv vJ Ku uK

p	Values
	Lt tL Ms sM Nr rN Oq qO Pp pP Qo oQ Rn nR Sm mS Tl lT Uk kU Vj jV Wi iW Xh hX Yg gY Zf fZ
2	0x 2w 4v 6u 8t Bo Dn Fm Hl Jk Lj Ni Ph Rg

p	Values
	zS Tf xT vU Ve tV rW Xd pX nY Zc lZ
3	0p 3o 6n 9m Bj Ei Hh Kg Nf Qe Td Wc xX uY Zb rZ
4	0l 4k 8j Dg Hf Le Pd Tc Xb xZ
5	3i 8h Bf Ge Ld

p	Values
	Qc Vb
6	0h 6g Be Hd Nc Tb Za
7	6f Dd Kc Rb Ya
8	0f 8e Hc Pb Xa
9	3e Ec Nb Wa
10	8d Bc Lb Va
11	4d Jb Ua
12	0d Hb Ta
13	9c Fb Sa
14	6c Db Ra
15	3c Bb Qa
16	0c

p	Values
	Pa
17	Oa
18	Na
19	Ma
20	8b La
21	6b Ka
22	4b Ja
23	2b Ia
24	0b Ha
25	Ga
26	Fa
27	Ea
28	Da
29	Ca
30	Ba
31	Aa