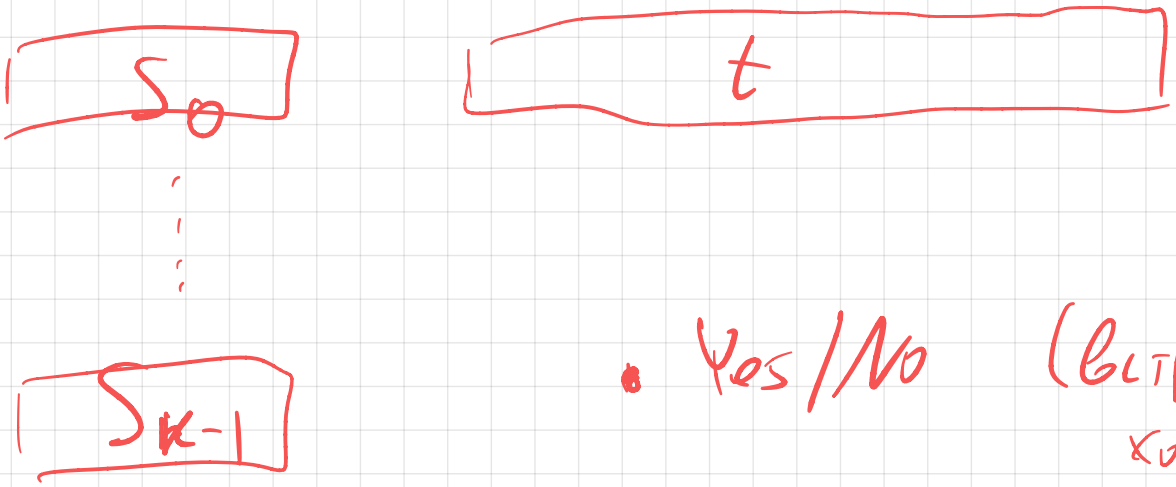
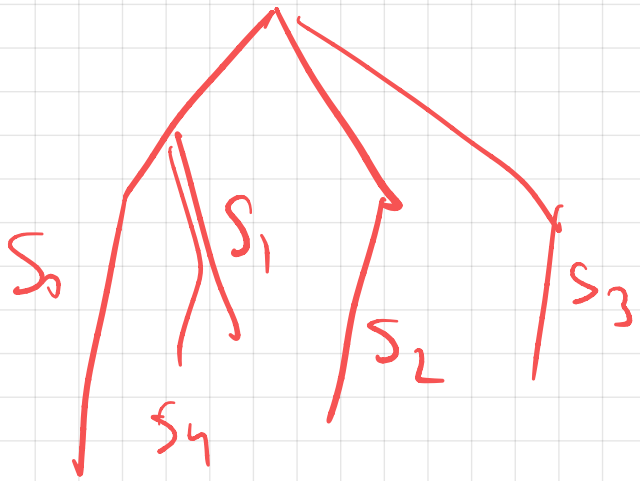
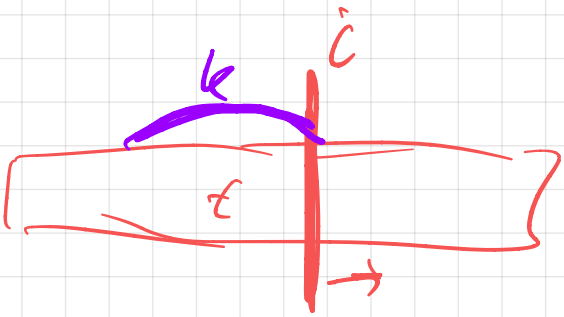
# Aho - Corasick

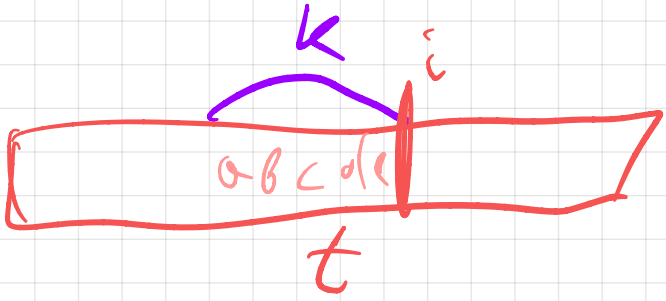


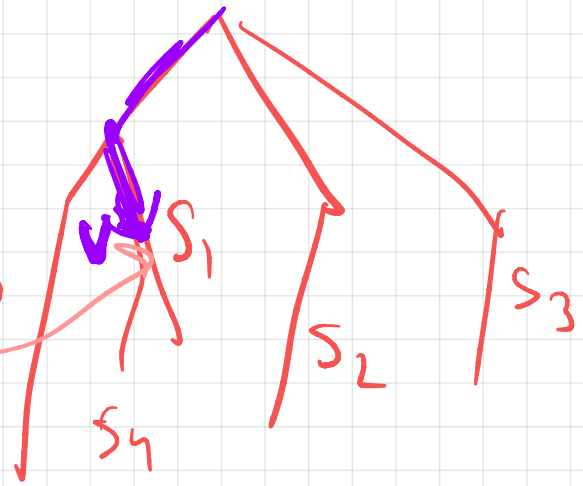

- Yes / No (встреч. ли хотя где-то)
- $i$ – конкретное вх-ие
- Найти все вх-ия во все места.
  - O(Вхождения)
  - Количество O(1)





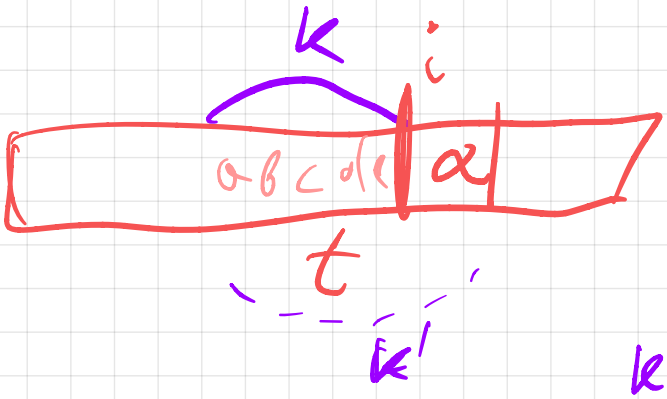


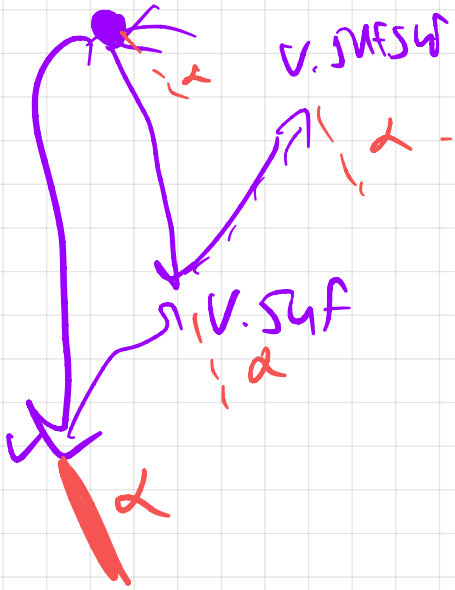



MAX k

$\Leftrightarrow$ самую глубокую в-ну в боре



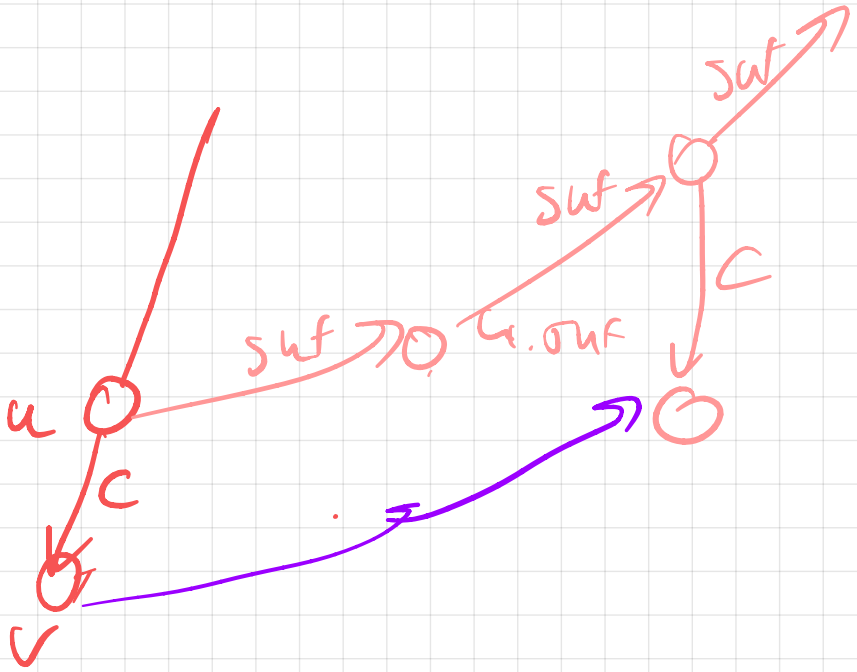

$k' \leq k+1$

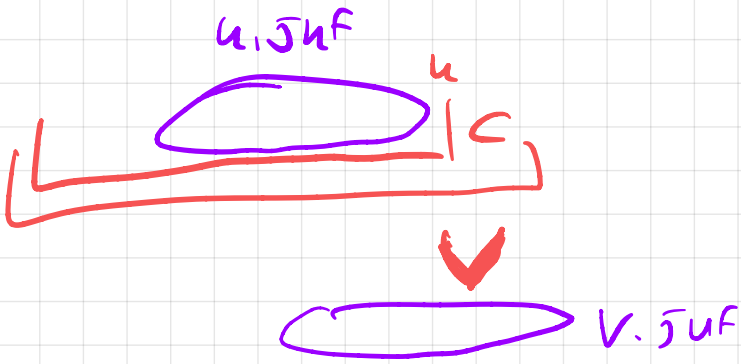
$\Rightarrow$ есть буква $\alpha$, перейдём по $\alpha$.

$\Rightarrow$ нет буквы ?

Def Суффиксная ссылка v.suf ведёт в в-ну, которая соотв. собств. суффиксу v, и среди таких самую глубокую

# Поиск суффссылок.



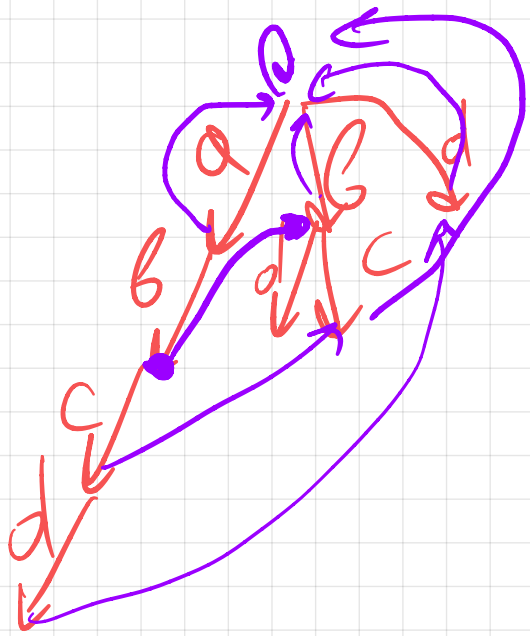




v.suf.parent – кандидат на u.suf

v.suf = ?

↖ u.suf = ?

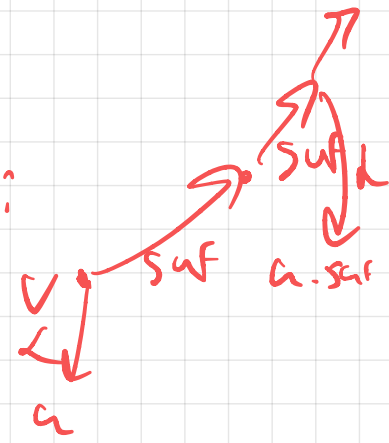
Э.П.




Псевдокод.

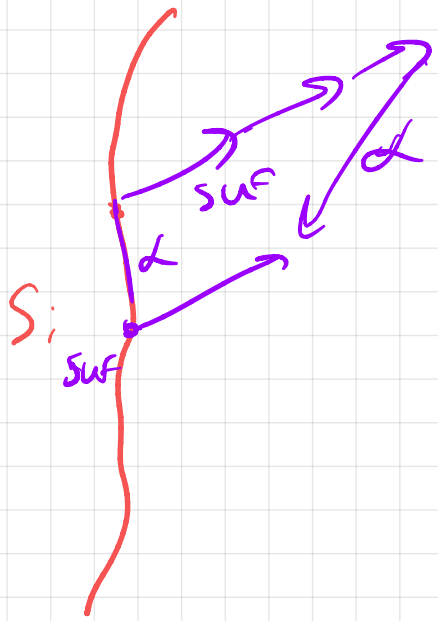
```
q = Queue()
[root.suf = root]
for [c, v] in root.go:
    v.suf = root
    q.push(v)

while q:
    v ← q.          // v.suf
    for [c, u] in v.go:
        u.suf = [?]
        q.push(u)
```

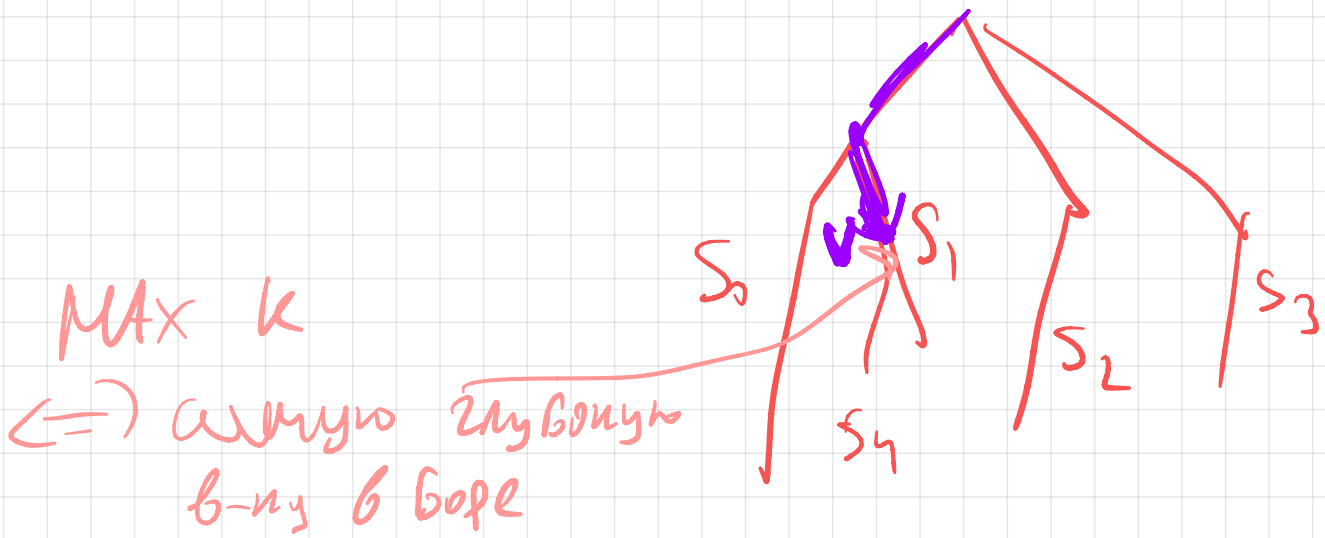



t = v.suf while(
t = t.suf

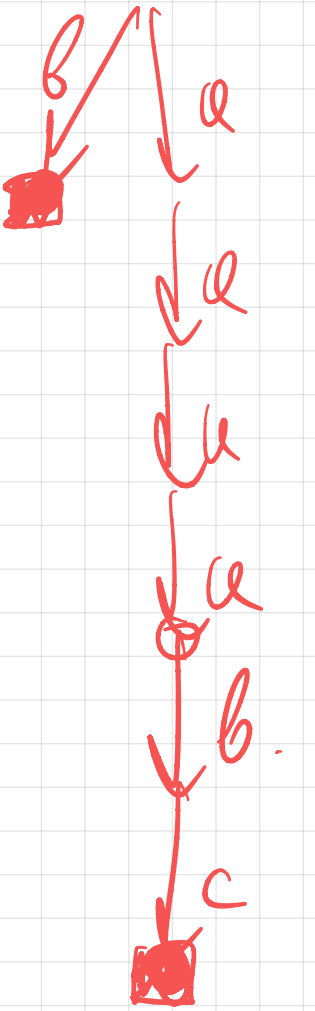
Ч1б : время работы : $O(\Sigma |S_i|)$



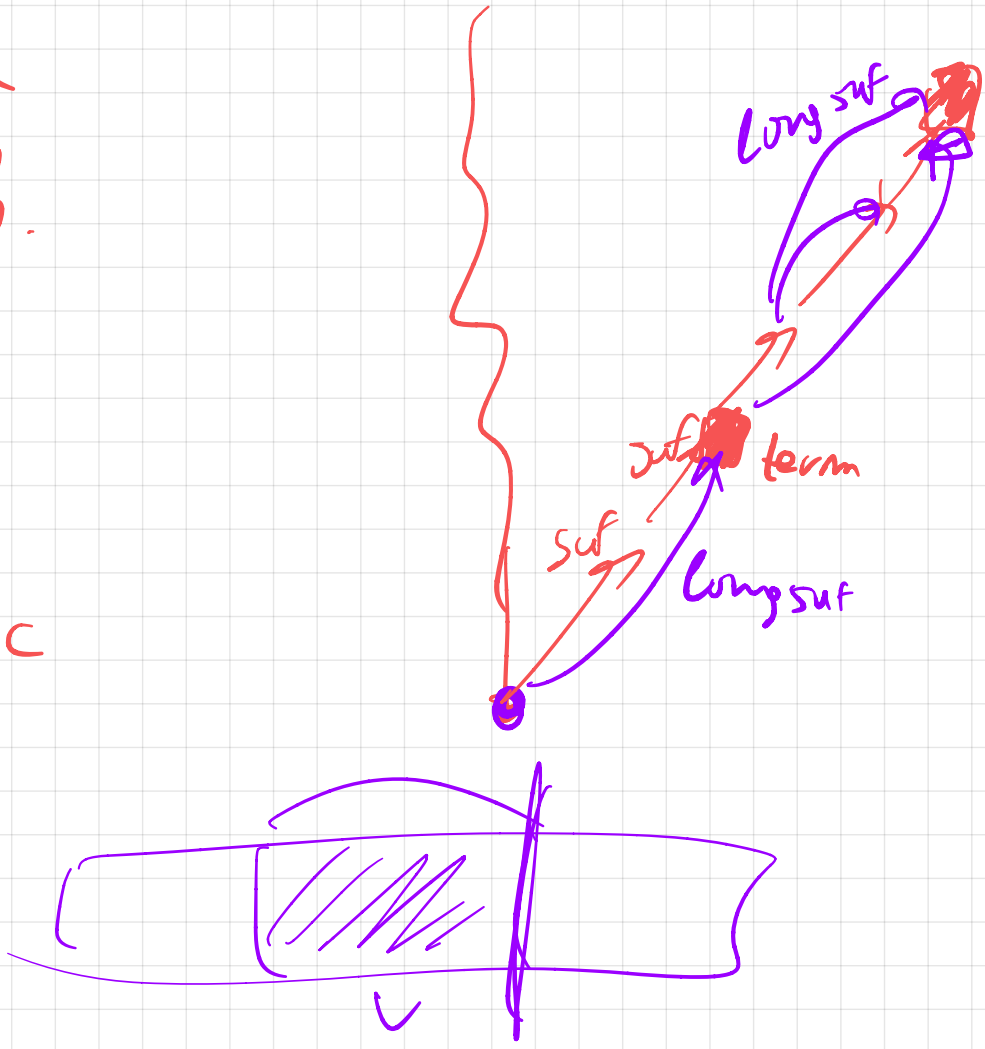

Поиск ответа




MAX k
$\Leftrightarrow$ самую глубокую в-ну в боре




aaaabc



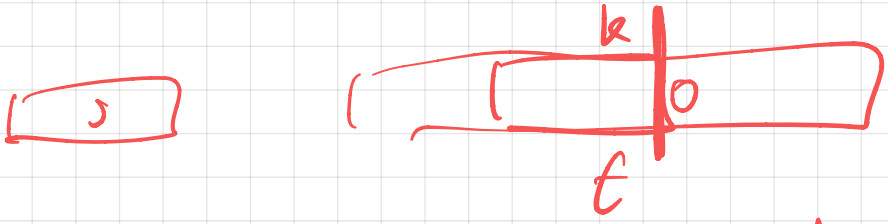




Yes/No: | $|V.term| = V.suf\_term$

Вхождение за $O(\text{вхождение})$: | $Long\ suf$

# Ахо-Корасик (автомат)




Построение:

$k \to k+1$

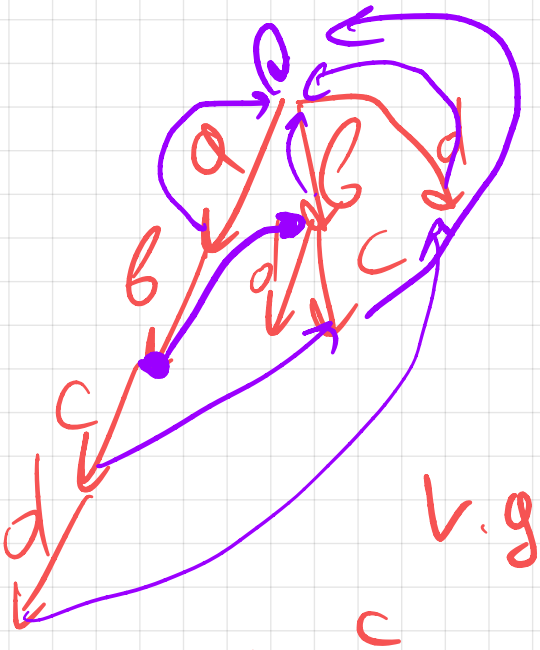
$k' \to k'+1$

$k'' \to k''+1$

$O(\Sigma|S| \cdot |\Sigma|)$

$O(S\Sigma_{бора} \cdot |\Sigma|)$

$k = go[k][\alpha]$.




for c in t:

$v = v.go[c]$.

$$v.go[c] = \begin{cases} u, & \text{если уже был} \\ & \text{такой переход} \\ v.suf.go[c] & \end{cases}$$

$u \to v$ $\quad u.suf = v.suf.go[c]$.

# Хеш-Таблица

dict / unordered_map / HashMap

```
a = dict()
a[10] = 245
a[10]
> 245
del a[10]
```

O(1)

Массив

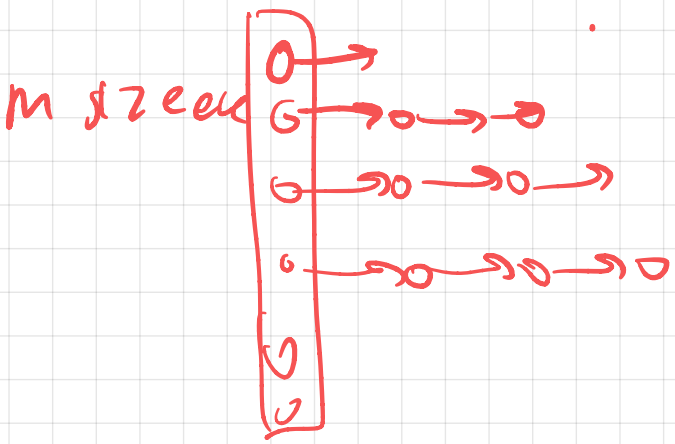
a[15] = 25

O(1)

# Separate Chaining

(с закрытой адресацией)

$$h: \underbrace{U}_{\text{ключей}} \to [0; M) \quad M \le 2^{64}$$




$$h_m: U \to [0; m)$$

$$h_m(k) = h(k) \,\%\, m.$$

```
get(k):
    i = h(k)
    for (k', v') in lists[i]:
        if k' == k:
            return v'
    return None
```

put (k, v)

$\alpha := n/m$ – ср-яя нагрузка.

$T_{get} = O(\alpha + 1)$

$T_{put} = O(\alpha + 1)$,

в предп. это амр случайный

$h \xleftarrow{random} U(H)$

$((ax + b) \bmod p) \bmod m$

$a \in \{1 .. p-1\}$

$b \in \{0 .. p-1\}$

$\alpha \le Const \Rightarrow T = O(1)$

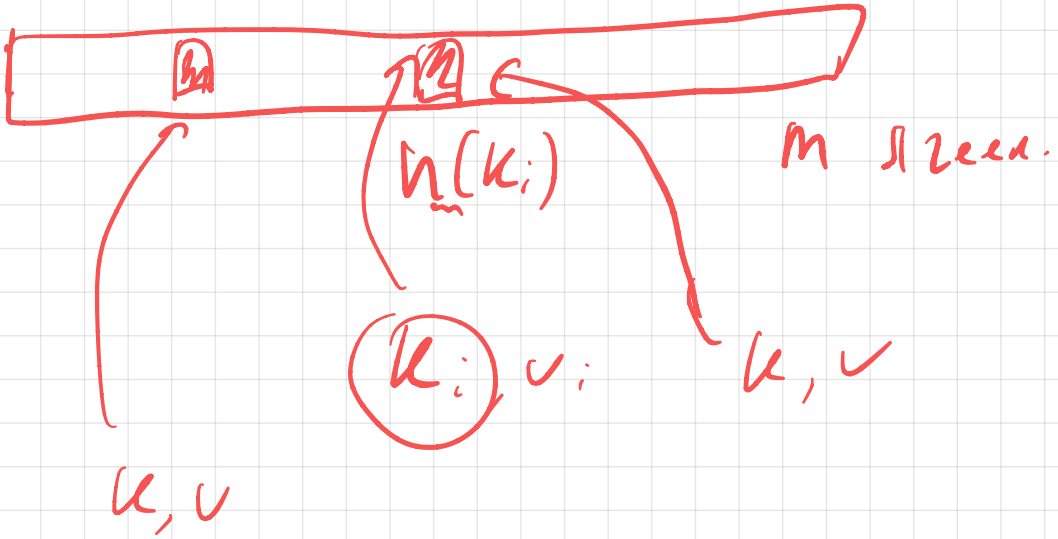
$\alpha > Const$:

↳ $m *= 2$, rehash $O(n)$

$\Rightarrow T = O(1)$ в среднем

на 1 вектор.

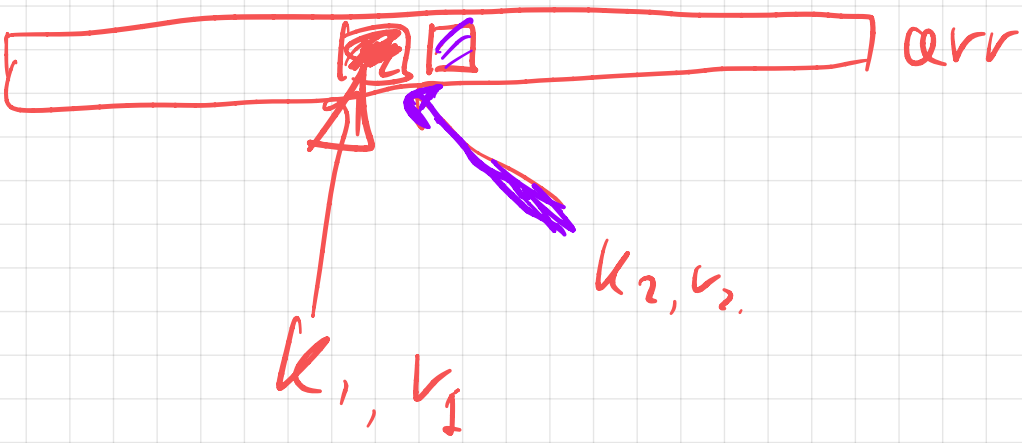
# Open Addressing




Порядок дней рождения


$\sqrt{m}$ испытаний

$P_{\text{вер конф.}} \simeq \frac{1}{2}$



```
get(k):
    i = h(k)
    while arr[i].k ≠ None &
          arr[i].k ≠ k:
        ++i
    if arr[i].k = None:
        return None
    return arr[i].v
```

$$\alpha = n/m$$

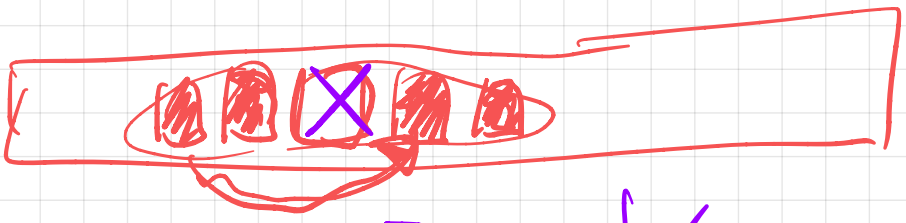
$$\Theta(1/\alpha)$$

$$\alpha \leq \frac{1}{2} \Rightarrow T = O(1)$$

(& ug. mupe)

$$\alpha > \frac{1}{2} \Rightarrow \text{rehash}$$

Удаление:



[key, $Val_{deleted}$]

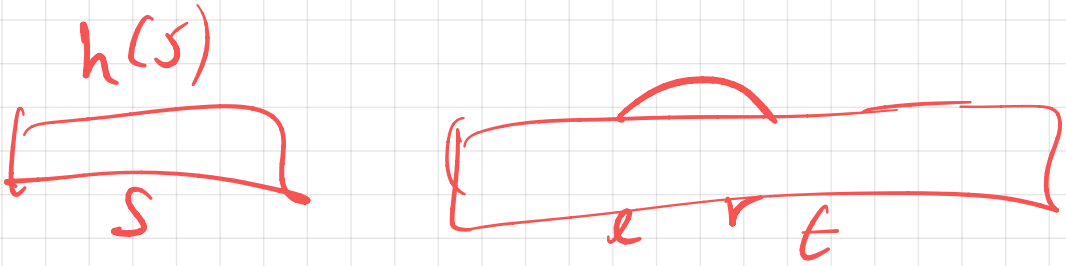
Коллизии

Хеширование Спрос.

$h: \Sigma^* \to$ Конечное множество

$h(s_1) \neq h(s_2) \Rightarrow s_1 \neq s_2$

$h(s_1) = h(s_2) \Rightarrow P[s_1 = s_2] \geq 1 - \varepsilon$




```
for i = 0.. |t| - |s| + 1
    if h(t_i ... t_{i+1+|s|}) ==
    == h(s):
        print (вхождение)
```

# Полиномиальный хеш

$$h(s_0 \dots s_{n-1}) = (p^{n-1} s_0 + p^{n-2} s_1 + \dots + p^0 s_{n-1}) \bmod m$$

$p$, $m$ — параметры хеша

$t_0 \dots t_{n-1}$

$h_i = h(t_0 \dots t_{i-1})$ (rolling hash)

$h_{i+1} = h(t_0 \dots t_i)$

$h_{i+1} = (p h_i + t_i) \bmod m$

Вычисляем хеш за каждый префикс.

hash(l..r)


$$h_{r+1} = (p^r s_0 + \dots + p^0 s_r) \bmod m$$

$$h_l = (p^{l-1} s_0 + \dots + p^0 s_{l-1}) \bmod m$$

$$h(s_0 \ldots s_{r-1}) = (h_{r+1} - h_0 p^{r+1-0}) \% m$$