$$n^d = O(d^n) \qquad d > 1$$

- $100^{100^{100}} = O(1)$
- $\log^a n = O(n^b) \qquad a, b > 0$

  $\log_2^{1000} n = O(n^{0.001})$

  $\log_{20}^{10} n = \Theta(\log^{10} n)$

$$\log_a b = \frac{\ln b}{\ln a} = \frac{\log_2 b}{\log_2 a}$$

$$\log_{20} n = \log_2 n / \log_2 20$$

- $n^a = O(b^n) \qquad a > 0, \; b > 1$

$S(n) \leq n \log n \leq O(n^2)$

$$\sqrt{n} = n^{1/2}$$

$$n^c = 2^{\log n \cdot c} \qquad \log\log n < \sqrt{\log n} < n$$

$$2^{\sqrt{\log n}} \qquad \log n = 2^{\log\log n}$$

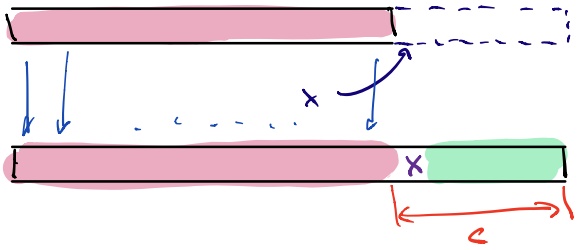
Структури данних

1. Масив

- Нашиб фикс. /огр. размера

Достуи, добавление в конец - удаление за $O(1)$
Удаление с сохр. порядка - $O(n)$

- Расширяемый массив с аддитивной схемой



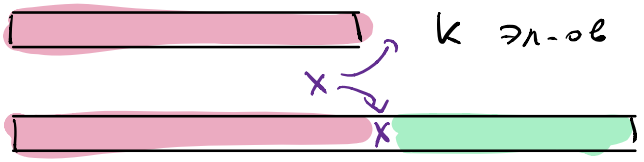
выделение

Пусть изначально масив пустой

$$\#\text{ копирований} = 0 + c + 2c + 3c + \dots + (n-c) = \frac{n-c}{2}\cdot\left(\frac{n}{c}-1\right) = \Theta(n^2)$$

накладные расходы

- расширяемый массив
с мультипликат. ск. аллокации



$$\#\text{ копирований} = \lfloor n/\alpha \rfloor + \lfloor n/\alpha^2 \rfloor + \lfloor n/\alpha^3 \rfloor + \dots + c \leq$$

$\leq \frac{n}{\alpha-1} + const$

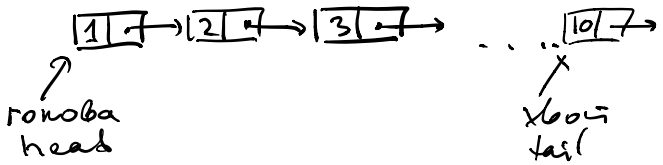
$$\leq \sum_{k=1}^{\infty} \lfloor n/\alpha^k \rfloor \leq \sum_{k=1}^{\infty} n/\alpha^k = n \cdot \frac{1}{1-1/\alpha} = \beta$$

$= \Theta(n)$

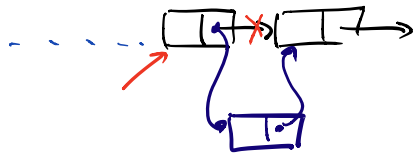
При $\alpha = 2$, $\beta = 1$.

2. Списки

– Односв. список



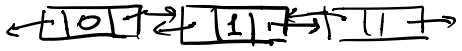
- удалять / добавлять в начало $O(1)$
- добавлять эл-т в середину после какого-то эл.ом



$O(1)$

- добавлять эл-т в конец

– двусвязн. список



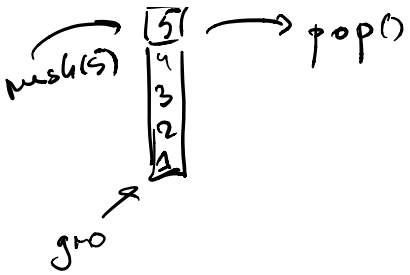
+ удаление из конца за $O(1)$

Абстрактные типы данных (АТД)

– стек

– pop() вытащить эл. LIFO
– push(x) добавить эл-т last in - first out

—

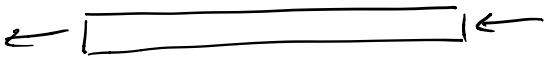


Реализации:
- на массиве
- на списке

- очередь

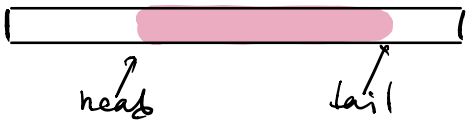
FIFO

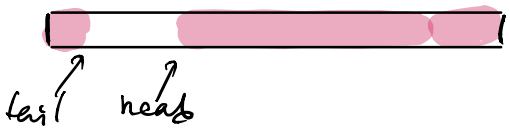
first in - first out



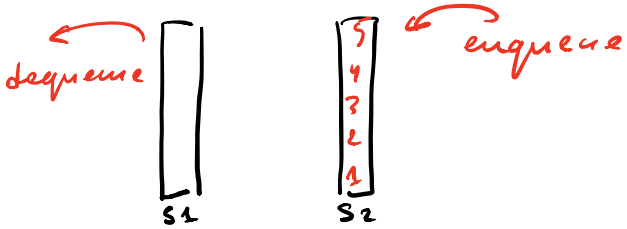
- enqueue(x) - добавить
- dequeue() - вытащить

Реализации:
- двусвязный список
- односвязный список
- массив: замкнутый массив (кольцевой буффер)





- Реализация на двух стеках



Амортизационный анализ

- Метод предоплаты

Всего $4 \cdot n$ монеток

Поэтому на n операций потратится $\leq 4n$ монеток $\Rightarrow O(n)$

$\Rightarrow$ Амортизированная сложность 1 операции $\leq 4n / n = 4 = O(1)$

- Метод потенциалов

c_i - стоимость i-ой операции

$\sum_{i=1}^{m} c_i$ - суммарная стоимость всех опер. $\leq m \cdot \max_i c_i$

$\tilde{c}_i$ - учётная стоимость операции

$\Phi : S \to \mathbb{R}_+$, $\Phi(S_0) = 0$

S_i - состояние структуры данных

$\Phi(S_i)$ - потенциал S_i - сост. структуры данных после i-ой операции

$\hat{c}_i = c_i + \Phi(S_i) - \Phi(S_{i-1})$

$$\sum_{i=1}^{m} \hat{c}_i = \sum_{i=1}^{m} c_i + \underbrace{\Phi(S_m) - \Phi(S_0)}_{\geq 0} \Rightarrow$$

$$\sum_{i=1}^{m} c_i = \sum_{i=1}^{m} \hat{c}_i \underbrace{- \Phi(S_m) + \Phi(S_0)}_{\leq 0} \leq \sum_{i=1}^{n} \tilde{c}_i$$

$$\leq m \cdot \max_i \tilde{c}_i$$

Примеры:

- Очередь на двух стеках

enqueue $c_i = 1$

dequeue $\begin{cases} c_i = 1 & \text{if } S_1.\text{size}() > 0 \\ c_i = 1 + S_2.\text{size}() & \text{otherwise} \end{cases}$

$$\tilde{c}_i = c_i + \Delta\Phi = O(1)$$

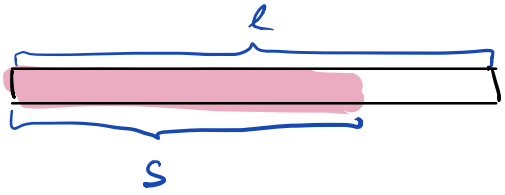
$$\Phi(S) = S_2.\text{size}()$$

enqueue $\tilde{c}_i = c_i + \Delta\Phi = 1 + 1 = 2$

dequeue $\begin{cases} \tilde{c}_i = 1 + 0 = 1 \\ \tilde{c}_i = 1 + S_2.\text{size}() - S_2.\text{size}() = 1 \end{cases}$

- Расширяемый массив с мультипликативной аллокацией

Состояние массива: s и ℓ



$\alpha = 2$

push_back: $\begin{cases} c_i = 1, & s < \ell \\ c_i = 1 + s, & s = \ell \end{cases}$

$$\Phi(S_i) = 2s - \ell$$

$$\begin{cases} \hat{c}_i = c_i + \Phi(S_i) - \Phi(S_{i-1}) = & s < \ell \\ \quad = 1 + (2(s+1) - \ell) - (2s - \ell) = 1 + 2 = 3 & \\ \hat{c}_i = 1 + s + (2(s+1) - 2\ell) - (2s - \ell) & s = \ell \\ \quad = 1 + s + 2 - \ell = 3 & \end{cases}$$