

Introduction

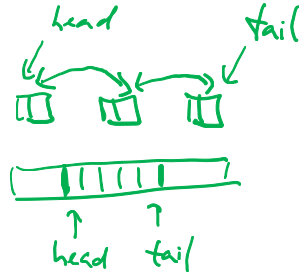
DS: data + operations (queries / updates)

"black box" interface

implementation \Rightarrow complexity of operations

static (only queries) / dynamic

EX: • QUEUE: ENQUEUE(x) $O(1)$ implementations:
 DEQUEUE $O(1)$ \Leftarrow linked list
 ISEMPTY $O(1)$ array



cyclically indexed

• STACK: PUSH(x), POP, ISEMPTY (similar)

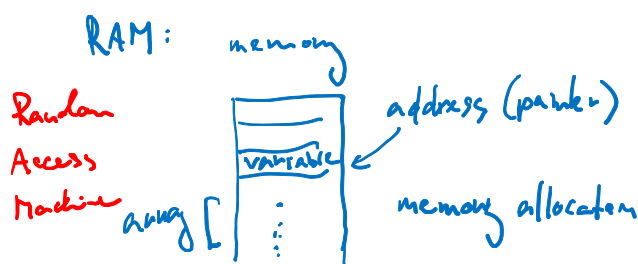
• SET:	INSERT(x)	$O(n)$	$O(n)$	$O(n)$	$O(\log n)$	$O(1)$	average complexities
$S \subseteq U$	DELETE(x)	$O(n)$	$O(n)$	$O(n)$	$O(\log n)$	$O(1)$	
	FIND(x)	$O(n)$	$O(n)$	$O(\log n)$	$O(\log n)$	$O(1)$	
	BUILD(x_1, \dots, x_n)	$O(n)$	$O(n)$	$O(n \log n)$	$O(n \log n)$	$O(n)$	
		linked list	array	sorted array	BST balanced	hash table	

• DICTIONARY: (key, value) (similar to SET)

• MULTISSET: DICTIONARY for (key, # occurrences)

• ORDERED SET:	MIN/MAX	$O(n)$	$O(n)$	$O(1)$	$O(\log n)$	$O(n)$	PRED(x) := max {y ∈ S y < x}
	PRED(x)	$O(n)$	$O(n)$	$O(\log n)$	$O(\log n)$	$O(n)$	
	SUCC(x)	$O(n)$	$O(n)$	$O(\log n)$	$O(\log n)$	$O(n)$	
		linked list	array	sorted array	BST balanced	hash table	

Model of computation



standard arithmetic + logic instructions (incl. comparison)

assumption: all integers used $\leq \text{poly}(\text{len}(I), \text{max}(I))$

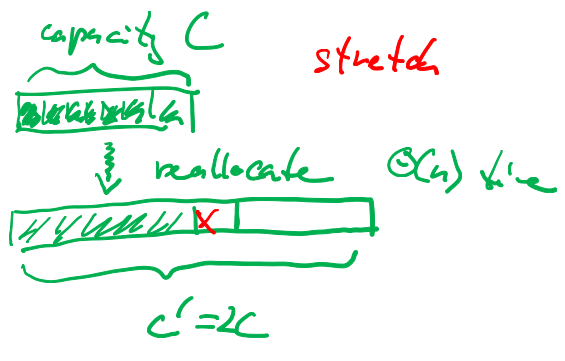
time = # instructions used

space = # cells used (incl. "spaces")

Amortized analysis

Aggregation method

Ex: stretchable array: APPEND(x)
 $\Theta(n)$ worst case time

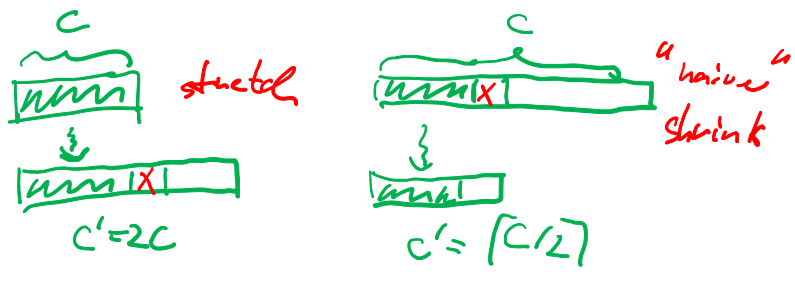


n items \Rightarrow final capacity $2^k, 2^{k-1} < n \leq 2^k$

(aggregated) total time = $2^0 + 2^1 + \dots + 2^{k-1} = 2^k - 1 = \Theta(n) \Rightarrow \Theta(1)$ amortized time for reallocations

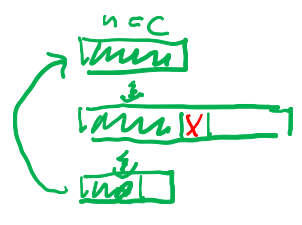
Accounting method

Ex: flexible array: APPEND(x)
 (stack) REMOVE LAST

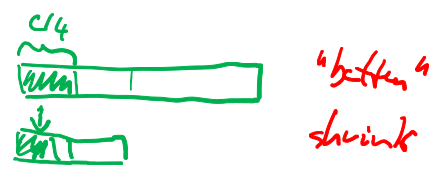


"naive" shrink: $C' = \lfloor C/2 \rfloor$ if $n < C/2$
 for $n = C + \text{APPEND}, \text{REMOVE LAST } 2x, \text{APPEND}$
 $\Rightarrow \Theta(n)$ amortized time

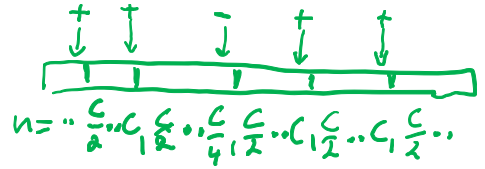
does not work:



"better" shrink: $C' = \lfloor C/4 \rfloor$ if $n < C/4$



n operations, divided into blocks by reallocations:



+ stretch, - shrink

⚡: After stretch/shrink we have $n = C/2 \pm 1$

\Rightarrow each block contains at least $C/4$ operations

reallocation cost $\Theta(C)$ accounted to operations in its block \Rightarrow

$\Theta(1)$ amortized time