Tampereen yliopisto

Amortized performance and std::vector's memory management

COMP.CS.300 Data structures and algorithms 1

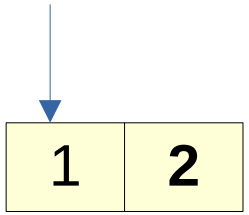
Matti Rintala (matti.rintala@tuni.fi)

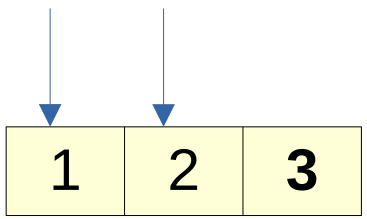
STL vector memory management

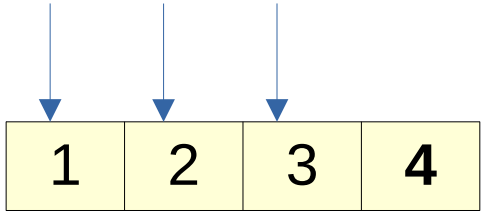
- Vector reserves a continuous memory block for its elements
- What to do when it needs more space?

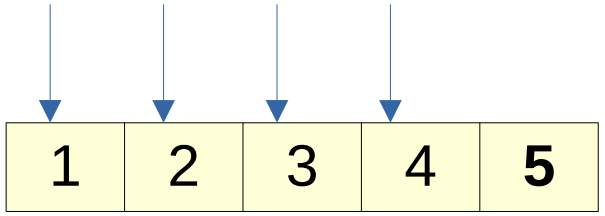
1

- Attempt 1: reserve new memory block with as much more space as you need, copy old elements there


1
2

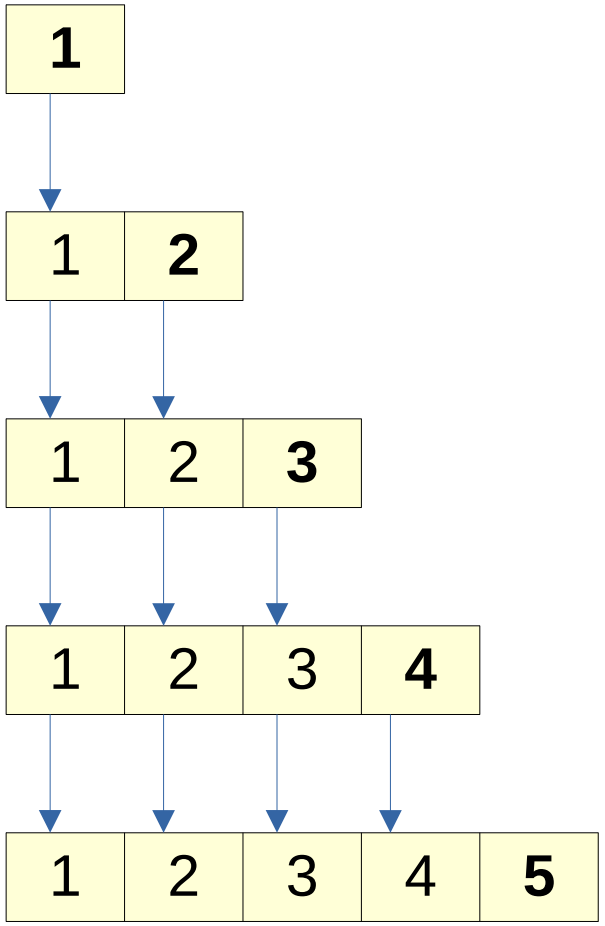

1
2
3


1
2
3
4


1
2
3
4
5

STL vector memory management

- Vector reserves a continuous memory block for its elements
- What to do when it needs more space?
- Attempt 1: reserve new memory block with as much more space as you need, copy old elements there

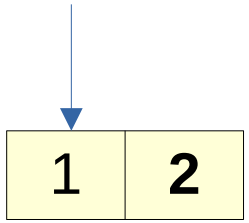


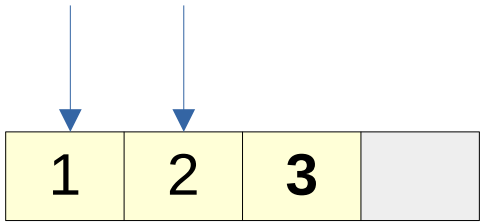
STL vector memory management

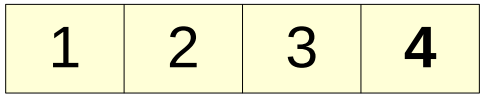
- Vector reserves a continuous memory block for its elements
- What to do when it needs more space?

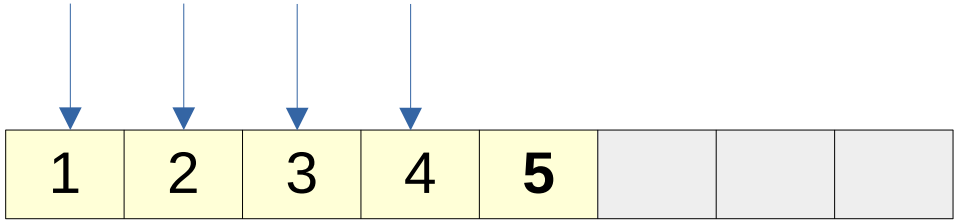
1

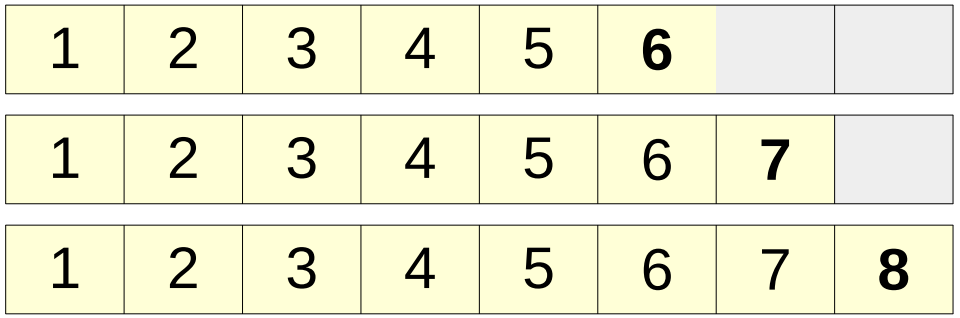
- Attempt 2: reserve **twice as large** memory block, copy old elements there
- Note: Some memory remains unused!


1
2


1
2
3

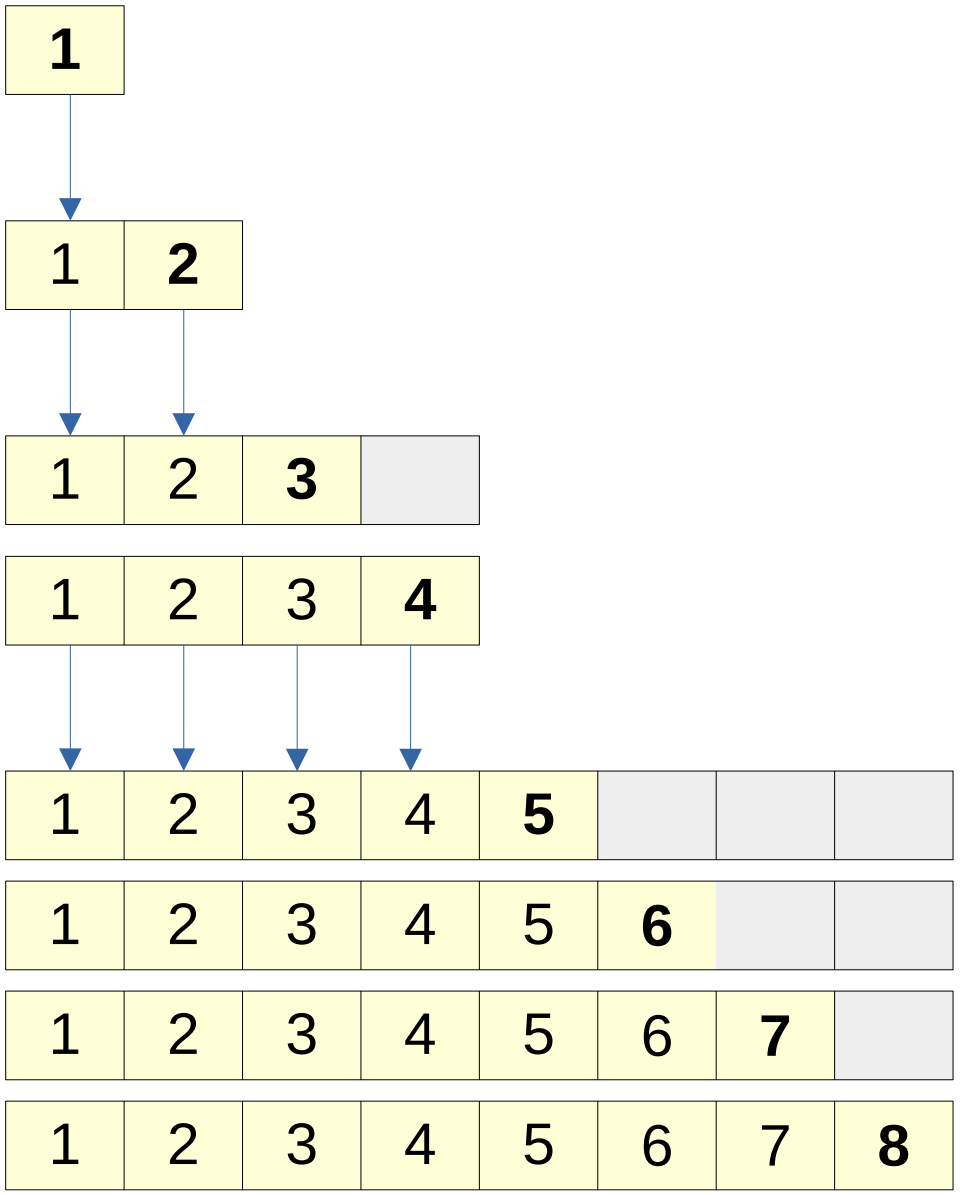

1
2
3
4


1
2
3
4
5


1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8

STL vector memory management

- Vector reserves a continuous memory block for its elements
- What to do when it needs more space?
- Attempt 2: reserve **twice as large** memory block, copy old elements there
- Note: Some memory remains unused!



Amortized tehokkuus

- Counts average performance of *operation sequences*
- Cost of expensive rare operation can be spread evenly on cheap operations
- E.g., append an element to vector:
  - Individual insertion can be linear (gets rarer and rarer)
  - Insertions are still *amortized* constant time (on average)

std::vector

- STL vector contains operations for tweaking memory management
- vec.reserve(n): Reserves *memory* at least for n elements, elements are still not (yet) added
- vec.capacity(): Maximum number of elements without new memory allocation
- vec.shrink_to_fit(): Move elements to memory block that is just the right size
- (vec.erase() *does not* free memory!)