向量

可扩充向量：分摊

02-B2

...在他的心理上，他总以为北平是天底下最可靠的大城，不管有什么灾难，到三个月必定灾消难满，而后诸事大吉。北平的灾难恰似一个人免不了有些头疼脑热，过几天自然会好了的。

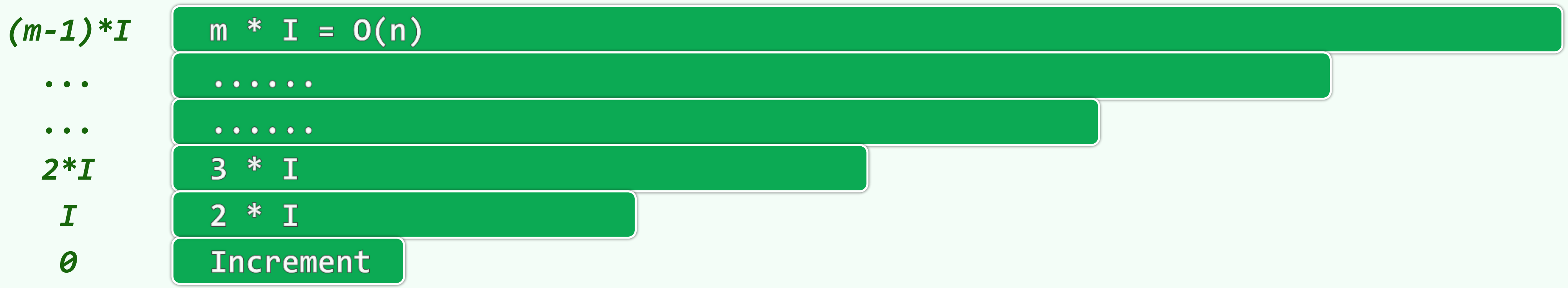
邓俊辉
deng@tsinghua.edu.cn

容量递增策略

- `T* oldElem = _elem; _elem = new T[ _capacity += INCREMENT ];` //追加固定增量
- 最坏情况：在初始容量0的空向量中，连续插入n = m*I >> 2个元素...
- 于是，在第1、I + 1、2I + 1、3I + 1、...次插入时，都需扩容
- 即便不计申请空间操作，各次扩容过程中复制原向量的时间成本依次为

  $0, I, 2I, \ldots, (m-1)*I$ //算术级数

  总体耗时 = $I * (m-1) * m/2 = \mathcal{O}(n^2)$，每次（insert/remove）操作的**分摊成本**为$\mathcal{O}(n)$



容量加倍策略

❖ `T* oldElem = _elem; _elem = new T[ _capacity <<= 1 ];` //容量加倍

❖ 最坏情况：在初始容量1的**满**向量中，连续插入n = 2^m >> 2个元素...

❖ 于是，在第1、2、4、8、16、...次插入时都需扩容

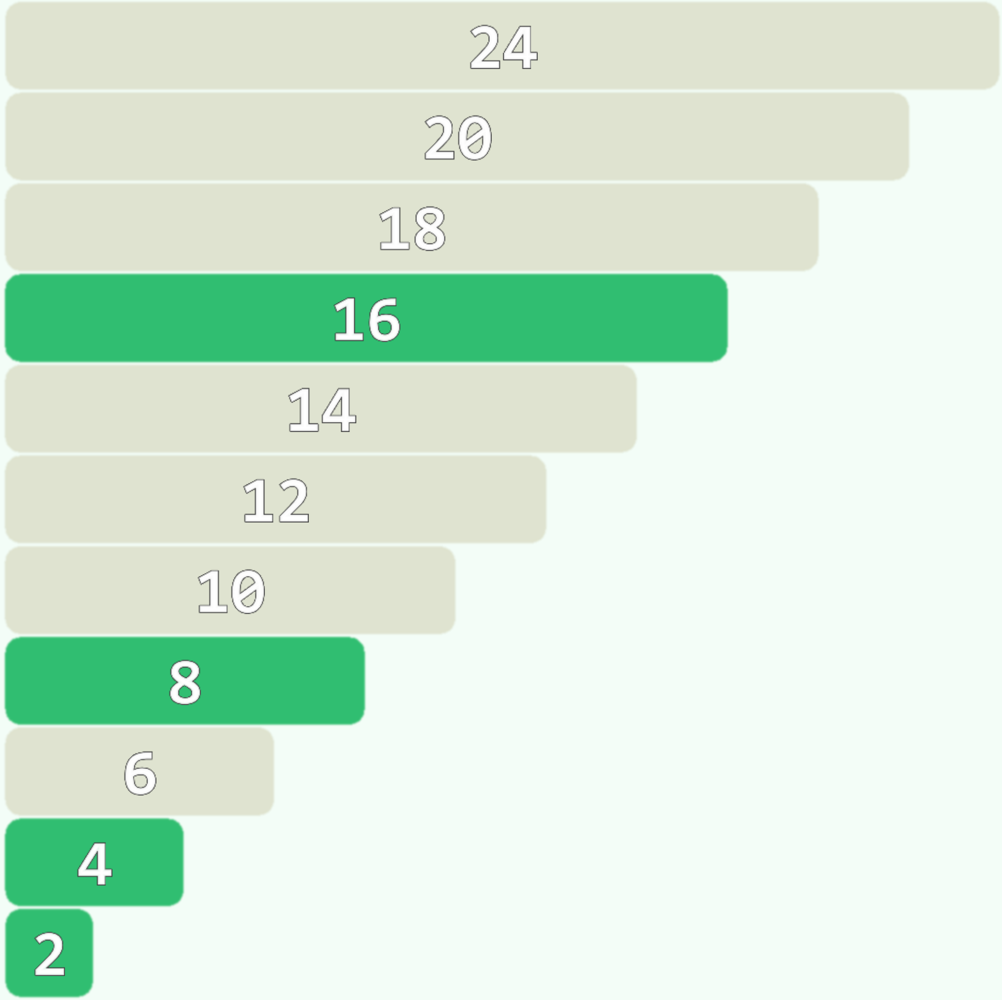
❖ 各次扩容过程中复制原向量的时间成本依次为

1, 2, 4, 8, ..., 2^m = n //几何级数

总体耗时 = $\mathcal{O}$(n)，每次（insert/remove）操作的**分摊成本**为$\mathcal{O}$(1)

对比

	递增策略	倍增策略
累计 增容时间	$\mathcal{O}(n^2)$	$\mathcal{O}(n)$
分摊 增容时间	$\mathcal{O}(n)$	$\mathcal{O}(1)$
装填因子	$\approx$ 100%	$>$ 50%



平均分析 vs. 分摊分析

- 平均（average complexity）：根据各种操作出现概率的分布，将对应的成本加权平均
  - 各种可能的操作，作为独立事件分别考查
  - 割裂了操作之间的相关性和连贯性
  - 往往不能准确地评判数据结构和算法的真实性能
- 分摊（amortized complexity）：连续实施的足够多次操作，所需总体成本摊还至单次操作
  - 从实际可行的角度，对一系列操作做整体的考量
  - 更加忠实地刻画了可能出现的操作序列
  - 更为精准地评判数据结构和算法的真实性能
- 后面将看到更多、更复杂的例子