

向量

可扩充向量：分摊

.....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

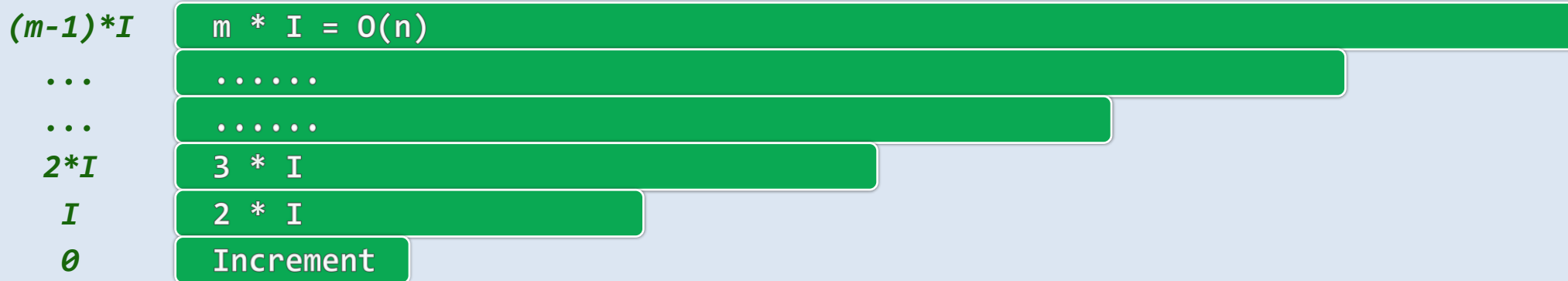
deng@tsinghua.edu.cn

# 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 $\theta$ 的空向量中，连续插入 $n = m \cdot I \gg 2$ 个元素，而无删除操作

❖ 于是，在第  $1, I + 1, 2I + 1, 3I + 1, 4I + 1, \dots$  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

$0, I, 2I, 3I, 4I, \dots (m - 1) \cdot I$  //算术级数

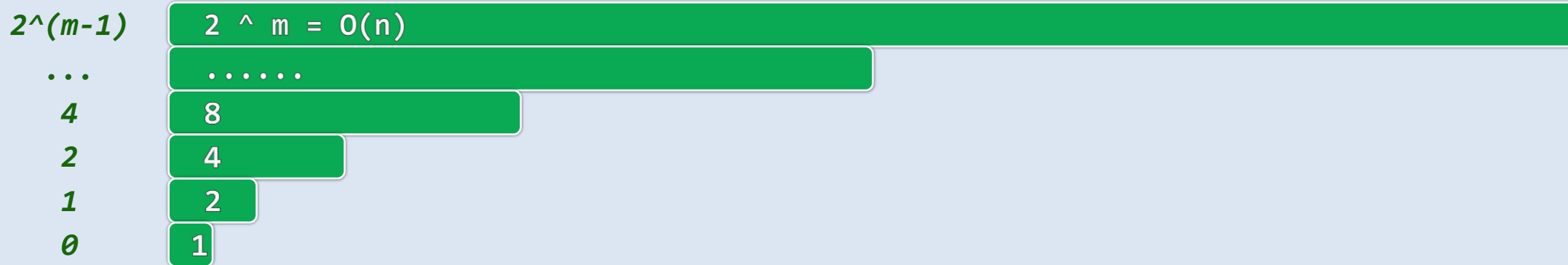
总体耗时 =  $O(n^2)$ ，每次 (insert/remove) 操作的分摊成本为  $O(n)$

# 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  $n = 2^m \gg 2$  个元素，而无删除操作

❖ 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

$1, 2, 4, 8, 16, \dots, 2^{m-1}, 2^m = n$  //几何级数

总体耗时 =  $O(n)$  , 每次 (insert/remove) 操作的分摊成本为  $O(1)$

# 对比



|                 | 递增策略            | 倍增策略     |
|-----------------|-----------------|----------|
| 累计扩容时间          | $O(n^2)$        | $O(n)$   |
| 分摊扩容时间          | $O(n)$          | $O(1)$   |
| 空间利用率<br>(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