

02-A-5 复制 & 02-B-1 可拓展向量

#数据结构邓神

复制

具体操作分为两步

- 1: 开辟空间
- 2: 复制



基于赋值的构造

```
template <typename T>
void Vector<T>::copyFrom(T* const A, Rank lo, Rank hi){
    _elem = new T[_capacity = 2 * (hi - lo)]; //分配空间
    _size = 0; // 规模清空
    while (lo < hi){ // 赋值2
        _elem[_size++] = A[lo++];
    }
}
```

可拓展向量

静态空间管理（效率低下）

静态空间管理

❖ 开辟内部数组 `_elem[]` 并使用一段地址连续的物理空间

`_capacity` : 总容量

`_size` : 当前的实际规模 `n`

`_elem`

`_size`

`_capacity`

❖ 若采用静态空间管理策略，容量 `_capacity` 固定，则有明显的不足

1. **上溢** (overflow) : `_elem[]` 不足以存放所有元素

尽管此时系统仍有足够的空间

2. **下溢** (underflow) : `_elem[]` 中的元素寥寥无几

装填因子 (load factor) $\lambda = \text{_size} / \text{_capacity} \ll 50\%$

❖ 更糟糕的是，一般的应用环境中难以准确**预测**空间的需求量

是否有方法使得向量可以根据使用的需求动态调整空间呢？同时保持高的效率？