

## 02-A-3 操作实例 & 02-A-4 构造与析构

#数据结构邓神

### ADT操作实例

ADT操作实例					
操作	输出	向量组成 (自左向右)	操作	输出	向量组成 (自左向右)
初始化			disordered()	3	4 3 7 4 9 6
insert(0, 9)		9	find(9)	4	4 3 7 4 9 6
insert(0, 4)		4 9	find(5)	-1	4 3 7 4 9 6
insert(1, 5)		4 5 9	sort()		3 4 4 6 7 9
put(1, 2)		4 2 9	disordered()	0	3 4 4 6 7 9
get(2)	9	4 2 9	search(1)	-1	3 4 4 6 7 9
insert(3, 6)		4 2 9 6	search(4)	2	3 4 4 6 7 9
insert(1, 7)		4 7 2 9 6	search(8)	4	3 4 4 6 7 9
remove(2)	2	4 7 9 6	search(9)	5	3 4 4 6 7 9
insert(1, 3)		4 3 7 9 6	search(10)	5	3 4 4 6 7 9
insert(3, 4)		4 3 7 4 9 6	uniquify()		3 4 6 7 9
size()	6	4 3 7 4 9 6	search(9)	4	3 4 6 7 9

### Vector 模版类

#### Vector模板类

```
typedef int Rank; //秩
#define DEFAULT_CAPACITY 3 //默认初始容量 (实际应用中可设置为更大)
template <typename T> class Vector { //向量模板类
private: Rank _size; int _capacity; T* _elem; //规模、容量、数据区
protected:
    /* ... 内部函数 */
public:
    /* ... 构造函数 */
    /* ... 析构函数 */
    /* ... 只读接口 */
    /* ... 可写接口 */
    /* ... 遍历接口 */
};
```

data structures & algorithms (Fall 2012), Tsinghua University

### 构造和析构

```
Vector(int c = DEFAULT_CAPACITY){ // 默认创建空数组
    _elem = new T[_capacity = c];
    _size = 0;
}
Vector(T const * A, Rank lo, Rank hi){ // 数组区间复制
    copyFrom(A, lo, hi);
}
Vector(Vector<T> const& V, Rank lo, Rank hi){ // 数组区间复制
    copyFrom(V._elem, lo, hi);
}
Vector(Vector<T> const& V){ //。数组整体复制
    copyFrom(V._elem, 0, V._size);
}
Vector(){
    delete [] _elem;
}
```