02-C-1 无序向量 & 02-C-2 元素访问

#数据结构邓神

模版 类

```
template <typename T>  // 定义一个模版类
class Vector{
  ...
}

Vector<int> // Vector of Integer
Vector<BinTree> // Vector of BinaryTree == Forest
```

无序向量

元素访问

似乎并不是什么问题，通过V.get(r) 和 V.put(r,e) 接口已经可以读写向量

似乎在形式上不够简洁直观
但就便捷性而言，远不如数组元素的访问方式 : A[r] // 可否借用的下标的访问方式?

可以！为此需要重载下标操作符号"[]"

```
template <typename T> // 0 <= r < _size
T & Vector<T>:: operater[](Rank r) const {
  return _elem[r];
}

// 注意这边的操作需要定义为 T* _elem 这样的指针数组才可以使用
// 如果定义的是普通的静态数组则需要改成
T & Vector<T>:: operater[](Rank r) const {
  return (T&)(_elem[r]); // 也就是需要强制转换为 &
}
```

为了便于讲解，这里采用了简单的方式来处理意外和错误（例如：入口参数越界等）
实际应用中，应该采用更加严格的方式