02 Efficient Dynamic Array

jjcao

Mostly Based on the assignment of Professor Ligang Liu

Problem

- 现在实现的类基本满足一个“动态数组”的要求，用户能很方便的使用它。
-
- 但是，这个动态数组的使用也有问题，比如，每当数组的元素个数发生变化时，总是要发生内存的释放和申请的操作，这个使得操作效率上非常低效，并且很容易产生内存碎片。想象一下如果用户是不断用push_back()函数来增加10000个元素，程序是如何的运行？
-
- 如何能改进它呢？

Solution

- 一个想法就是预先多分配一些内存，然后在相当一段时间内，当数组的元素个数发生变化时，不用重新申请内存空间，数组也能继续使用。这样就提高了效率。

- 对于机器算法中，“时间”和“空间”的矛盾总是存在的：存储多一点，运行就快些；存储少了，运行就慢些。在这里就体现了，这样处理是用空间换时间。虽然在内存空间上是“浪费”了部分的空间，但是在很长一段时间里，只要空间没有发生变化，运行的时间复杂度就恒定的。只有当现有的空间不够用时，才发生内存的重新申请和释放操作。最简单的增长方法就是增长一倍，即乘以2，这样内存分配的大小就是2的幂次方。

- 因此，我们加上一个m_capacity数据，用以存储所申请到的内存空间大小，而m_size才真正记录这个数组的元素个数。

The solution seems easy. But how to organize related function consistently?

```
class EfficientArray
{
private:
    double* m_data; // the pointer to the array memory
    int     m_size; // the size of the array
    int     m_capacity; // the max memory of the array
```

Functions to be modified:

- **3 constructors**
- **"=" operator**
- **resize -> reserve:**
- **erase: reduce memory is no longer needed**
- **Insert:**
- **push_back**

Comparison

```
#include <boost/timer.hpp>

BArray b;  EArray e;
boost::timer t; // start timing
int asize(59999);
for ( int i = 0; i < asize; ++i)   b.push_back(1);
double elapsed_time_b = t.elapsed();

t.restart();
for ( int i = 0; i < asize; ++i)   e.push_back(1);
double elapsed_time_e = t.elapsed();

double tim = elapsed_time_b - elapsed_time_e; // 18.24 in my PC

std::cout << "EfficientArray is " << tim  << " seconds faster than BasicArray!" <<
std::endl;
```

Boost

"...one of the most highly regarded and expertly designed C++ library projects in the world."

— Herb Sutter and Andrei Alexandrescu, C++ Coding Standards

- Introduction: http://www.boost.org/
- Download: http://www.boostpro.com/download/
- 不必下载所有的libraries，通常源代码就够了
- Some libraries I favored:
  - Boost.Threads
  - boost_graph
  - Boost.Program_options

A template project with requirements

1. 实现hw02_EfficientDynamicArray_Template要求的高效动态数组
   - 实现Insertion Sort算法

a. 完成满足上述接口的动态数组(Dynamic array)的程序，递交工程文件(*.vcxproj, *.vcxproj.user, (旧的版本*.dsp)) 和源文件(*.h, *.cpp)；
b. 工程目录中的debug目录删除掉，其他文件压缩打包发给我，参照我们提供的code template。

评分标准	分数
程序成功运行（通过所有assert）	60
正确使用new/delete，不出现内存泄漏	10
需要遵循基本的编程规范和风格；	20
按时完整的提交项目文件，不包含无用文件	10

抄袭则 平均分配 成绩 = ?/n

hw02_EfficientDynamicArray_Template in http://www.jjcao.net/c.html

通过完成该作业需达到的目标：

- 体会类的**接口确定**了，其**实现可以不一样**：用户只跟类（对象）的接口打交道，具体是通过哪些数据来实现的是不关心的
- 进一步熟悉和体会C++类的封装特性；
- 仍使用以前的测试代码，做更多充分的**测试**，体会到处理各种**极端情况**需要更多的代码和调试时间
- 严格遵循基本的**编程规范和风格**；