

```
A *pa = new A(); //操作符  
operator new (011014FBh) //函数  
malloc() //c语言中的malloc()函数  
A::A() //有构造函数就调用
```

```
delete pa;  
A::~A(); //析构函数  
operator delete();  
free() //c语言中的free()函数
```

`A *pa = new A[3]();`

`0x01275218`

`0x0127521c`

记录分配了多少个字节  
的数字 (4字节)

Debug模式下可能会有  
些调试信息  
占用一些字节 (30-60字节)

4个字节记录  
数组大小

分配出去的  
10字节

其他一些必要信息, 一些为  
提高效率进行边界调整的  
字节填充 (十-几十字节)

尾信息(4字节)用于回收  
内存时当做内存尾标记

//一: new内存分配细节探秘  
//我们注意到, 一块内存的回收, 影响范围很广, 远远不是10个字节, 而是一大片。  
//分配内存这个事, 绝不是简单的分配出去4个字节, 而是在这4个字节周围, 编译器做了很多处理, 比如记录分配出去的字节数等等;

//new, delete(malloc, free)内存没有看上去那么简单, 他们的工作内部是很复杂的;