

# 常见编程技巧

## 内存池

当我们需要动态分配内存的时候，频繁使用 **new/malloc** 会占用大量的时间和空间，甚至生成大量的内存碎片从而降低程序的性能，可能会使原本正确的程序 **TLE/MLE**。

这时候我们就需要使用到「内存池」这种技巧：在真正使用内存之前，先申请分配一定大小的内存作为备用，当需要动态分配时则直接从备用内存中分配一块即可。

当然在大多数 **OI** 题当中，我们可以预先算出需要使用到的最大内存并一次性申请分配。

如申请动态分配**32**位有符号整数数组的代码：

```
inline int* newarr(int sz) {
    static int pool[maxn], *allocp = pool;
    return allocp += sz, allocp - sz;
}
```

线段树动态开点的代码：

```
inline Node* newnode() {
    static Node pool[maxn << 1], *allocp = pool - 1;
    return ++allocp;
}
```

From:  
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:  
<https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:namespace:sereinin:%E7%9F%A5%E8%AF%86%E7%82%B9&rev=1589552324>

Last update: 2020/05/15 22:18