# 单调队列：

所谓单调队列，就是队里元素是单调递增或递减的队列,一旦入队元素使得队列不单调，就把队尾元素删除直到满足单调性后再插入那个新元素，这个实际是栈的性质，本来队列是尾入头出，而不能删除尾部。所以有的地方叫做单调栈，但因为队列支持从队头出队的功能(在滑动窗口问题有用到)，所以习惯叫单调队列，它确实不是真正的队列，它同时具有栈和队列的性质。

编码上，c++有deque的双端队列可用，也可以用来手动实现。(有人说c++库慢，但之后实测一点不慢)。但是最快的是手动实现的数组双端队列。这么做在算法竞赛中只要开够了内存(2倍于峰值)，就不需要释放。

## 模型一,找字典序：

给定一个序列，找字典序最小的子序列。从头到尾构建单调队列即可

## 模型二,滑动窗口问题：

所谓滑动窗口就是给一个长度是n的数列A,对于数列每一段长度是m的区间，都要求得一个最值，**单调递减队列**可以用来求滑动窗口的**最大值(这里习惯队尾在前，队头在后，所以单调递减队列是维护最大值)**。其算法复杂度是O(n)。这是正招，如果用堆或线段树等结构，复杂度是n\*logn，单调队列的一个用途是利用其滑动窗口最值优化动态规划问题的时间复杂度。

例如poj2823，以滑动最大值为例

用单调递减队列记录最大值，构建单调队列整个过程队列头部的数所在位置和队列尾部所在位置一定不能超过m,如果超过了就要把队列头部出队(保证了求长度是m的区间内最值)。如何知道队列每个元素位置，在队列额外维护一个属性即可。

先把A序列序列前m个元素按照单调队列规则入队，同时保证队列长度，处理完前m个数，队尾就是最大值(很好理解，单调队列尾永远是最大的)，之后按顺序每次入队A的一个元素A[i]，每次输出队列尾部，就是区间[i-m+1,i]最大值，同时一定保证队列长度.这里要注意边界问题，因为是区间最大值，所以单调队列出队时，相等元素也要出队，这样可以更新所存的位置变量，使其更靠后

## 模型三,递增递减子序列等dp问题优化：

这个模板主要是二分+单调队列优化最长递增子序列问题。动态规划分区有讲

但是hdu6319这道题考了滑动区间的递增子序列问题。

## 例题：

Hdu6319 hdu3410 [**poj2823**](http://www.baidu.com/link?url=Lui4ONJgONje7bPxbsQUVbXZfIxng7srlwBJUNYm7rjC_gb8lVSnDBL5-UPfCm0usz4Fcz1kNpe16asWflkx_ZG_pAYsnpFxHLbAu2tWIXcl2xaFHPBfNj5ztalJFsXL)