灵知的太阳解题报告

---samjia2000

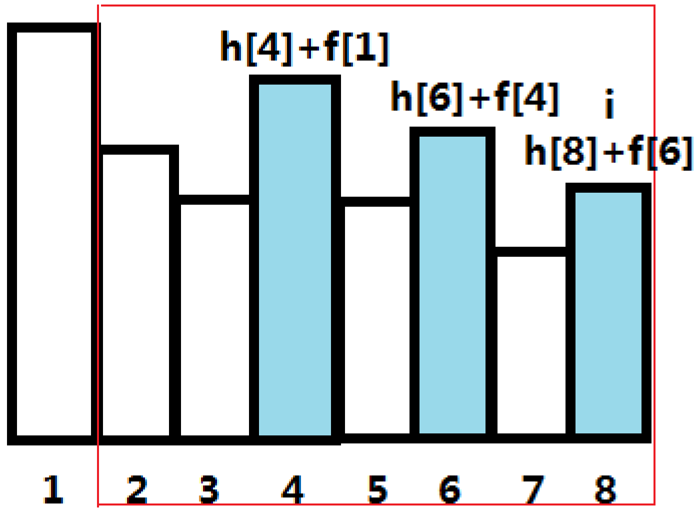
首先来想最简单的动态规划，通过第一个数的限制，我们可以得到每个位置的状态可以从哪些位置转移过来，得到如下形式的式子：

$$f(i) = max_{j=l[i]}^{i-1}(s[j+1][i] + f[j])$$

其中 s[i][j]表示从 i 到 j 的最大值，l[i]表示 i 位置最多向左走到哪里。

然后套路一波，用个单调递减的单调队列，每次将队头不合法的依次踢掉，然后我们来想想，这样我们只是得到了可以转移过来的状态集，那么如何维护答案呢？

观察下图：



现在我们的 i 在位置 8 上，红色框表示 i 向左最多到达的位置（即 2 与 8 的 p[i]是一样的）可以看出，除了队头之外，其他的在单调队列里的元素的贡献都是从前面一个位置的 f 转移而来的（因为 f 是单调不减的），而队头的则是依赖于 l[i]的。

假设当前的队列为:$a[head..tail]$且$\forall head \le i < tail, a[i] < a[i+1]$

那么当前点的答案为:$\max(max_{j=head+1}^{tail}(h[a[j]] + f[a[j-1]]), h[a[head]] + f[l[i]-1])$

观察到每次我们移动一个点是有许多没有变的状态的，每个元素进一次出一次，于是我们用堆或者其他数据结构来维护队列中每个元素的贡献，对于队头特殊处理。

时间复杂度$O(n\,log_2 n)$