

# 知识精炼（四）



主讲人：邓哲也



## Codeforces 834D. The Bakery

一个序列有  $n$  个数，你要把他们分成  $k$  段。  
每一段的价值等于这一段中出现的数字种数。  
请你求出价值和的最大值。

$1 \leq n \leq 35000$ ,  $1 \leq k \leq \min(n, 50)$

样例输入：

7 2

1 3 3 1 4 4 4

答案： 5

## Codeforces 834D. The Bakery

经典的线形模型。

可以用  $f[i][j]$  来表示前  $i$  个数分成  $j$  段的最大价值。

转移：

$$f[i][j] = \max \{f[k][j - 1] + w(k + 1, i) \mid k < i\}$$

$w(k + 1, i)$  就是指  $[k+1, i]$  中出现的数字种类数。

暴力的转移需要  $O(n^2k)$

## Codeforces 834D. The Bakery

考虑我们是怎么统计一个区间  $[1, r]$  中有几种不同的数的？

我们可以记录每个数  $a[i]$  左边第一个和它相同的数的位置

$pre[i]$

考虑  $a[i]$  这个数对  $w$  的贡献。

$w(pre[i], i)$  中已经有 1 的贡献了。

但是对于  $pre[i] < j \leq i$ ,  $w(j, i)$  中没有算进  $a[i]$  的贡献，因为在  $a[i]$  出现之前， $[j, i-1]$  中没有出现过  $a[i]$ 。

因此  $a[i]$  出现要对  $[pre[i] + 1, i]$  中的  $w$  都加一。

## Codeforces 834D. The Bakery

$$f[i][j] = \max \{f[k][j - 1] + w(k + 1, i) \mid k < i\}$$

我们只要在第  $k$  个位置上维护  $f[k][j-1]+w(k+1, i)$

计算  $f[i][j]$  时，先对  $[pre[i], i-1]$  上的数加一（对  $w$  的贡献）

然后取最大值转移即可更新  $f[i][j]$ 。

时间复杂度为  $O(nk \log n)$

# BZOJ 2678 Bookshelf

```
#define ls (x << 1)
#define rs (x << 1 | 1)
void upd(int x) {
    seg[x] = max(seg[ls], seg[rs]);
}
void down(int x) {
    if(add[x]) {
        seg[ls] += add[x];
        seg[rs] += add[x];
        add[ls] += add[x];
        add[rs] += add[x];
        add[x] = 0;
    }
}
```

## BZOJ 2678 Bookshelf

```
void modify(int ql, int qr, ll v, int l, int r, int x) {
    if(ql <= l && r <= qr) {
        seg[x] += v;
        add[x] += v;
        return;
    }
    int mid = (l + r) >> 1;
    down(x);
    if(ql <= mid) modify(ql, qr, v, l, mid, ls);
    if(mid < qr) modify(ql, qr, v, mid + 1, r, rs);
    upd(x);
}
```

## BZOJ 2678 Bookshelf

```
11 ask(int ql, int qr, int l, int r, int x) {  
    if(ql <= l && r <= qr) return seg[x];  
    int mid = (l + r) >> 1;  
    ll ret = 0;  
    down(x);  
    if(ql <= mid) ret = max(ret, ask(ql, qr, l, mid, ls));  
    if(mid < qr) ret = max(ret, ask(ql, qr, mid + 1, r, rs));  
    return ret;  
}
```



# BZOJ 2678 Bookshelf

```
int main() {
    scanf("%d%d", &n, &k);
    for(int i = 1; i <= n; i++) {
        scanf("%d", &x);
        pre[i] = pos[x];
        pos[x] = i;
    }
    for(int i = 1; i <= n; i++) {
        f[i][1] = f[i - 1][1];
        if (!pre[i]) f[i][1]++;
    }
    for(int j = 2; j <= k; j++) {
        memset(seg, 0, sizeof(seg));
        memset(add, 0, sizeof(add));
    }
}
```

## BZOJ 2678 Bookshelf

```
        for(int i = 1;i <= n;i ++)  
            modify(i, i, f[i][j - 1], 1, n, 1);  
        for(int i = j;i <= n;i ++){  
            modify(pre[i], i - 1, 1, 1, n, 1);  
            f[i][j] = ask(j - 1, i - 1, 1, n, 1);  
        }  
    }  
    printf( "%d\n" , f[n][k]);  
    return 0;  
}
```

下节课再见