知识精炼（四）

主讲人：邓哲也

BZOJ 1786 配对

有一个长度为 n 的序列，每个数都是 1～K 中的整数。

现在有一些位置的数被遮住了，用 -1 表示。

你可以往这些位置填1～K 中的数，使得整个序列的逆序对数最小，求最小的逆序对数。

$N \leqslant 10000$, $K \leqslant 100$

BZOJ 1786 配对

首先来考虑填进去的数有什么特征。

可以证明填进去的数是单调上升的。

我们可以取出填进去的数的连续两个 a 和 b。

a 和 b 的中间是原数列，设中间序列为 L。

假设 a < b，那么会产生多少个逆序对呢？

对于 L 中小于 a 的数，每个都会产生 1 个逆序对

对于 L 中大于 b 的数，每个都会产生 1 个逆序对

也就是 count(L, <=a) + count(L, >= b)

BZOJ 1786 配对

交换 a 和 b，会产生多少个逆序对呢？

对于 L 中小于 b 的数，每个都会产生 1 个逆序对

对于 L 中大于 a 的数，每个都会产生 1 个逆序对

也就是 count(L, <=b) + count(L, >= a)

a 在前 b 在后： count(L, <=a) + count(L, >= b)

b 在前 a 在后： count(L, <=b) + count(L, >= a)

因为 $a \leqslant b$，一定是 a 在 b 前面更优。

BZOJ 1786 配对

因此我们得到了一个结论：填进去的序列是单调不降的。

设计状态：

f[i][j] 表示已经填到第 i 个数了，这位填了 j 产生的逆序对数。

填了 j 会产生多少个逆序对呢？

只需要考虑原序列产生的贡献，即：

A[1..i-1] 中大于 j 的个数。

A[i+1..n] 中小于 j 的个数。

我们可以预处理两个数组 g[i][j] 和 l[i][j]，分别表示这两个含义。

BZOJ 1786 配对

因此 f[i][j] = min{ f[i-1][k] + g[i][j] + l[i][j] }

最后再加上数列本身的逆序对数就行了。

BZOJ 1786 配对

```
for(int i = 1;i <= n;i ++){
	scanf( “%d” , &a[i]);
	if (a[i] != -1) p[++ m] = i;
}
for(int i = 2;i <= n;i ++){
	for(int j = 1;j <= k;j ++) g[i][j] = g[i  -
1][j];
	if (a[i  -  1] != -1)
	for(int j = 1;j < a[i  -  1];j ++) g[i][j] ++;
}
```

BZOJ 1786 配对

```
for(int i = n - 1;i;i --){
	for(int j = 1;j <= k;j ++) l[i][j] = l[i + 
1][j];
	if(a[i + 1] != -1)
		for(int j = a[i + 1] + 1;j <= k;j ++)
			l[i][j] ++;
}
for(int i = 1;i <= n;i ++)
	if(a[i] != -1) ans += g[i][a[i]];
```

BZOJ 1786 配对

```
memset(f, 63, sizeof(f));
for(int i = 1;i <= k;i ++)
        f[1][i] = g[p[1]][i] + l[p[1]][i];
for(int i = 2;i <= m;i ++)
        for(int j = 1;j <= k;j ++)
                for(int o = 1;o <= j;o ++)
                        f[i][j] = min(f[i][j], f[i - 1][o] + g[p[i]][j] +
l[p[i]][j]);
ans2 = 1e9;
for(int i = 1;i <= k;i ++) ans2 = min(ans2, f[m][i]);
printf("%d\n", (ans2 < 1e9) * ans2 + ans);
```

下节课再见