知识精炼

主讲人：邓哲也

POJ 2288 岛屿和桥

给定一个地图，地图中有许多岛屿，岛屿之间用桥连接。汉密尔顿路径是一条沿着桥访问每个岛屿一次且仅一次的路径。在地图中，每个岛屿还有一个正整数权值。如果某条汉密尔顿路径使得下面描述的权值取得最大，则称这条汉密尔顿路径为最好的三角汉密尔顿路径。

POJ 2288 岛屿和桥

假设有 n 个岛屿，令 V[i] 为岛屿 C[i] 的权值，一条汉密尔顿路径 C[1]，C[2]，…，C[n] 的值分为 3 部分之和：

第一部分：将路径中每个岛屿的权值累加起来；

第二部分：对路径中的每条边 (C[i]，C[i+1])，将乘积(V[i] * V[i+1]) 加起来

第三部分：当路径中连续的三个岛屿 C[i]，C[i+1]，C[i+2]形成三角形时，即 C[i] 与 C[i+2] 之间有一座桥时，将乘积 V[i] * V[i+1] * V[i+2] 加起来。

请计算最好的三角汉密尔顿路径的权值，以及最好的路径数目。

n <= 13

POJ 2288 岛屿和桥

注意到汉密尔顿回路没用通用的求解策略，一般只能用指数级做法来搜索。

对于这里的 $n <= 13$，我们可以用二进制来表示每个岛屿是否被经过。

同时为了计算权值，我们不仅需要知道当前路径经过了哪几个点，还要知道最近经过了那个点，和距离当前倒数第二个经过的点，用来算后两部分的权值。

POJ 2288 岛屿和桥

因此状态可以设置为三维数组：dp[state][last][last_two]

统计路径个数的：cnt[state][last][last_two]

状态转移自然是从小到大进行，每次枚举下一个加入路径的点 i。

更新状态：

state’ |= 1 << i

last’ = i

last_two’ = last

POJ 2288 岛屿和桥

```
dp[state’ ][last’ ][last_two’ ] = max {
        dp[state’ ][last’ ][last_two’ ],
        dp[state][last][last_two] + C[i] + V[i] * V[last] +
V[i] * V[last] * V[last_two] * has_edge(last_two, i) }
```

cnt 数组跟着一起更新。

如果上述两者相等，那么：

```
cnt[state’ ][last’ ][last_two’ ] += cnt[state][last][last_two]
```

POJ 2288 岛屿和桥

dp数组的状态分别是 2^n 维、n 维、n 维。

每次枚举 n 个点。

因此总共的时间复杂度为 $O(n^3 2^n)$.

n=13可以解决。

下节课再见