知识精炼（四）

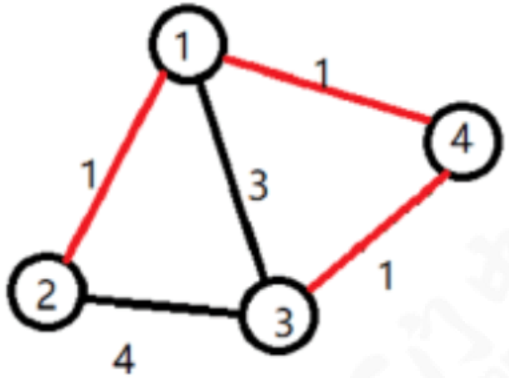
主讲人：邓哲也

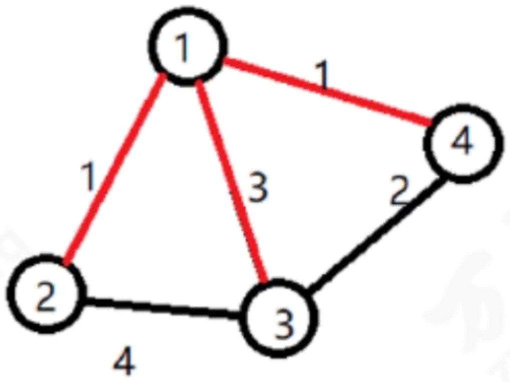
NOIP 2017 Day2 宝藏

简化版题目：

给定一个 n 个点 m 条边的图，请你求出一个有根树，满足每个点的深度和它到父节点的边权乘积之和最小。

$n \leqslant 12$，$m \leqslant 1000$





NOIP 2017 Day2 宝藏

考虑到点数只有12个，可以考虑状态压缩 DP。

用 s 表示当前加入的点集。

为了方便转移，我们不记录根是谁，而是直接去考虑深度。

也就是用 f[i][s] 表示当前的点集是 s，最深的点为 i。

然后我们去枚举 s 的补集的子集 t，把 t 都作为第 i+1 层加入 s。

NOIP 2017 Day2 宝藏

我们不用去考虑 t 里的点在这颗树中是否真的是第 i+1层

因为如果不是的话只可能小于i+1层，答案会更小。

那么一定存在一种转移顺序，考虑到这种更优的情况，也就

是先把这个点加入 s 集合。

NOIP 2017 Day2 宝藏

具体的操作是：

对于 s，枚举 t（s 的补集的子集），检查 t 里的点是否都和 s 里的点有连边，处理出每个点到 s 里的点的最短边。

设这些最短边边权之和为 v。

那么 f[i][s | t] = min(f[i][s | t], f[i - 1][s] + (i - 1) * v)

NOIP 2017 Day2 宝藏

时间复杂度分析：

s 一共有 2^n 个，s 的补集的子集一共有 3^n 个。

处理 t 里的每个点到 s 里的点的最短边，预处理时间复杂度 $O(n^2)$

验证 t 是否可行，时间复杂度 $O(n)$。

转移时对每个深度都要更新一次，时间复杂度$O(n)$

总时间复杂度就是 $O(n^2 2^n + n3^n)$，即 $O(n3^n)$

NOIP 2017 Day2 宝藏

```
memset(e, 63, sizeof(e));
scanf("%d%d", &n, &m);
for (int i = 1;i <= m;i ++){
    scanf("%d%d%d", &x, &y, &z);
    x --;
    y --;
    e[x][y] = e[y][x] = min(e[x][y], z);
}
for (int s = 0; s < (1 << n); s ++){
    for (int i = 0;i < n;i ++){
        v[i] = 0x3f3f3f3f;
        for (int j = 0;j < n;j ++) if (s >> j & 1)
            v[i] = min(v[i], e[i][j]);
    }
```

NOIP 2017 Day2 宝藏

```
    int c = ((1 << n) - 1) ^ s;
    for (int t = c; t; t = (t - 1) & c){
        int sum = 0;
        for(int i = 0;i < n;i ++) if (t >> i & 1){
            sum += v[i];
            if (sum >= 0x3f3f3f3f) break;
        }
        if (sum < 0x3f3f3f3f)
            for(int i = 1; i <= n;i ++)
                f[i][s | t] = min(f[i][s | t], f[i -
1][s] + (i - 1) * sum);
    }
}
```

NOIP 2017 Day2 宝藏

```
int ans = 0x3f3f3f3f;
for (int i = 1;i <= n;i ++)
    ans = min(ans, f[i][(1 << n) - 1]);
printf( “%d\n” , ans);
```

下节课再见