知识精炼（二）

主讲人：邓哲也

HDU 5593 ZYB’s Tree

给定一颗 n 个节点的树，每条边的边权为 1.

对于每个点，求出离这个点距离不超过 K 的点的个数。

$N \leqslant 500000$, $K \leqslant 10$

HDU 5593 ZYB’s Tree

先来考虑对于每个点 i，离 i 距离不超过 k 的子孙节点有多少个。

记为 f[i][k]

显然有 f[i][k] = sum(f[j][k - 1]) （j 是 i 的子节点）

HDU 5593 ZYB’ s Tree

稍微麻烦的一点就是离 i 距离不超过 k 的点可能不在 i 的子树中。

第一种可能：从兄弟节点转移过来

f[i][k] += f[j][k - 2] （j 是 i 的兄弟节点）

HDU 5593 ZYB’s Tree

稍微麻烦的一点就是离 i 距离不超过 k 的点可能不在 i 的子树中。记为 g[i][k]

第一种可能：从兄弟节点转移过来

g[i][k] += f[j][k - 2] （j 是 i 的兄弟节点）

记 sum[fa[i]] = sum(f[j][k - 2]) （j 是 fa[i] 的子节点）

这样就不用枚举 i 的兄弟节点，直接

g[i][k] += sum[fa[i]] - f[i][k - 2]

HDU 5593 ZYB’s Tree

第二种：从父亲节点转移过来

g[i][j] += g[fa[i]][j - 1]

这时一次 dfs 就不够了，因为计算 g 需要先计算 f。

f 是从孩子往父亲更新的。

g 是从父亲往孩子更新的。

而且更新 g 的时候需要 f 的值。

因此需要 dfs 两次。

HDU 5593 ZYB’s Tree

```
void dfs1(int u, int fa){
	for(int i =h[u]; i != -1; i = e[i].next){
		int v = e[i].v;
		if (v == fa) continue;
		dfs1(v, u);
		for(int j = 0;j <= K;j ++){
			if (j > 0) f[u][j] += f[v][j - 1];
			sum[u][j] += f[v][j];
		}
	}
}
```

HDU 5593 ZYB’ s Tree

```
void dfs2(int u, int fa){
	if (fa)
		for (int j = 1;j <= K;j ++)
			g[u][j] += g[fa][j - 1];
	for (int j = 2;j <= K;j ++)
		if (j >= 2 && fa)
			g[u][j] += sum[fa][j - 2] - f[u][j
- 2];
	for (int i = h[u]; i != -1;i = e[i].next)
		if (e[i].v != fa)
			dfs2(e[i].v, u);
}
```

HDU 5593 ZYB’s Tree

初始条件： f[i][0] = g[i][0] = 1

调用 dfs1(1, 0); dfs2(1, 0);

第 i 个点的答案就是 f[i][K] + g[i][K] - 1

时间复杂度 O(n)

下节课再见