

最近公共祖先—— Tarjan算法代码实现



主讲人：邓哲也



Tarjan算法代码实现

- 树的存储结构:
- **struct edge{**
- **int v, next;**
- **}e[M];**
- **int h[N], etot;**

- 采用邻接链表来存树，其中h[]是头数组，e[]是边数组

Tarjan算法代码实现

- 添加边
- `void add_edge(int x, int y) {`
- `e[etot].v = y;`
- `e[etot].next = h[x];`
- `h[x] = etot ++;`
- `}`
- `h[]`初始化为-1, `etot`初始化为0

Tarjan算法代码实现

- 询问也要按照邻接链表存起来
- **struct query {**
- **int v, next, ans;**
- **}q[M];**
- **int Q[N], qtot;**
- Q[]初始化为-1， qtot初始化为0

Tarjan算法代码实现

- 对于每一个询问 (u, v) ，调用 `add_query(u, v)`,
`add_query(v, u)`
- **`void add_query(int u, int v) {`**
- **`q[qtot].v = v;`**
- **`q[qtot].next = Q[u];`**
- **`qtot ++;`**
- **`}`**

Tarjan算法代码实现

- 维护一个并查集， $f[]$ 记录每个点的父亲，初始 $f[i]=i$
- `int find(int x) {`
- `if (x == f[x]) return x;`
- `else return f[x] = find(f[x]);`
- `}`
- `void merge(int v, int u){`
- `v = find(v);`
- `u = find(u);`
- `if (v != u) fa[v] = u;`
- `}`

Tarjan算法代码实现

- Tarjan算法主体，dfs的过程中合并父亲，回答询问。
- **void tarjan(int u, int fa) {**
- **for (int i = h[u]; i != -1; i = e[i].next) {**
- **if (e[i].v != fa) {**
- **tarjan(e[i].v, u);**
- **merge(e[i].v, u);**
- **vis[e[i].v] = 1;**
- **}**
- **}**
- **for (int i = Q[u]; i != -1; i = q[i].next)**
- **if (vis[q[i].v])**
- **q[i].ans = find(q[i].v);**
- **}**

下节课再见