

# 知识精炼（五）



主讲人：邓哲也



## Codeforces 893 F. Subtree Minimum Query

给定一颗树，树上的每个点有一个点权，每条边的边权都是 1。  
每次询问给出一对  $(x, k)$  点  $x$  的子树中，距离  $x$  小于等于  $k$  的所有点中最小值是多少。

$n \leq 100000$

## Codeforces 893 F. Subtree Minimum Query

设每个节点的深度为  $d[i]$

如果对  $x$  的子树里的节点，记录下深度  $= i$  时点权的最小值  $Min[i]$ 。

这样查询距离  $x$  不超过  $k$  的点，就是在  $Min[d[x] \dots d[x] + k]$  中查询最小值。

我们可以考虑用线段树来维护  $Min$  数组。

## Codeforces 893 F. Subtree Minimum Query

每个节点都对应一颗线段树。

但是节点  $x$  的线段树可以由  $x$  的孩子的线段树合并得到。

这里就可以用到线段树的合并操作。

最开始从叶子开始，线段树中只有一个值。

然后逐步合并上来。

最后只要存下每个节点对应的线段树的根节点，就可以回答询问了。

## Codeforces 893 F. Subtree Minimum Query

```
void dfs(int u, int fa) {  
    modify(d[u], a[u], 1, n, rt[u]);  
    for (int i = h[u]; i != -1; i = e[i].next) {  
        int v = e[i].v;  
        if (v == fa) continue;  
        d[v] = d[u] + 1;  
        dfs(v, u);  
        rt[u] = merge(rt[u], rt[v]);  
    }  
}
```

## Codeforces 893 F. Subtree Minimum Query

```
void update(int x) {  
    Min[x] = min(Min[lc[x]], Min[rc[x]]);  
}
```

## Codeforces 893 F. Subtree Minimum Query

```
void modify(int p, int v, int l, int r, int &x) {
    if (!x) x = ++ tot;
    if (l == r) {
        Min[x] = v;
        return;
    }
    int mid = (l + r) >> 1;
    if (p <= mid) modify(p, v, l, mid, lc[x]);
    else modify(p, v, mid + 1, r, rc[x]);
    update(x);
}
```

## Codeforces 893 F. Subtree Minimum Query

```
int merge(int x, int y){
    if(!x || !y) return x + y;
    int t = ++ tot;
    lc[t] = merge(lc[x], lc[y]);
    rc[t] = merge(rc[x], rc[y]);
    Min[t] = min(Min[x], Min[y]);
    return t;
}
```



## Codeforces 893 F. Subtree Minimum Query

对于询问  $x$   $k$ 。

只要在  $rt[x]$  上查询  $d[x] \dots \min(n, d[x] + k)$  的区间和即可。

时空复杂度都是：  $O(n \log n)$

下节课再见