

# 笛卡尔树

## 算法简介

一种二叉树，主要用于解决有关序列  $\text{RMQ}$  问题和一些矩阵维护问题，建树时间复杂度  $O(n)$

## 算法思想

笛卡尔树具有如下三条性质：

1. 对笛卡尔树中序遍历可以得到原序列
2. 笛卡尔树的点权符合堆性质
3. 任意区间的  $\text{RMQ}$  操作可以转换为笛卡尔树上两端点的  $\text{LCA}$  操作

接下来考虑建树，事实上可以按顺序插入序列，每次插入时由于插入结点下标最大，所以只能不停跳右子树。

找到合适结点，取代该结点位置，并将该结点所在子树变成插入结点的左子树。

发现只需要用一个单调栈维护一下从根结点起沿途所有右儿子构成的树链即可，时间复杂度  $O(n)$

## 代码模板

[洛谷p5854](#)

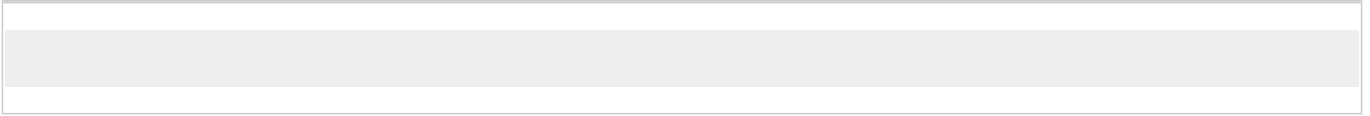
```
const int MAXN=1e7+5;
int Stack[MAXN],v[MAXN],ch[MAXN][2];
void build(int n){//小根堆的笛卡尔树
    int top=0,last=0;
    _rep(i,1,n){
        while(top&&v[i]<v[Stack[top]])top--;
        if(top)ch[Stack[top]][1]=i;
        if(top<last)ch[i][0]=Stack[top+1];
        Stack[++top]=i;
        last=top;
    }
}
```

## 算法练习

[SP3734](#)

## 题意

## 题解



From: <https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link: [https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal\\_string:jxm2001:%E7%AC%9B%E5%8D%A1%E5%B0%94%E6%A0%91&rev=1593924373](https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:legal_string:jxm2001:%E7%AC%9B%E5%8D%A1%E5%B0%94%E6%A0%91&rev=1593924373) 

Last update: 2020/07/05 12:46