01-E-7 MAX2

#数据结构邓神

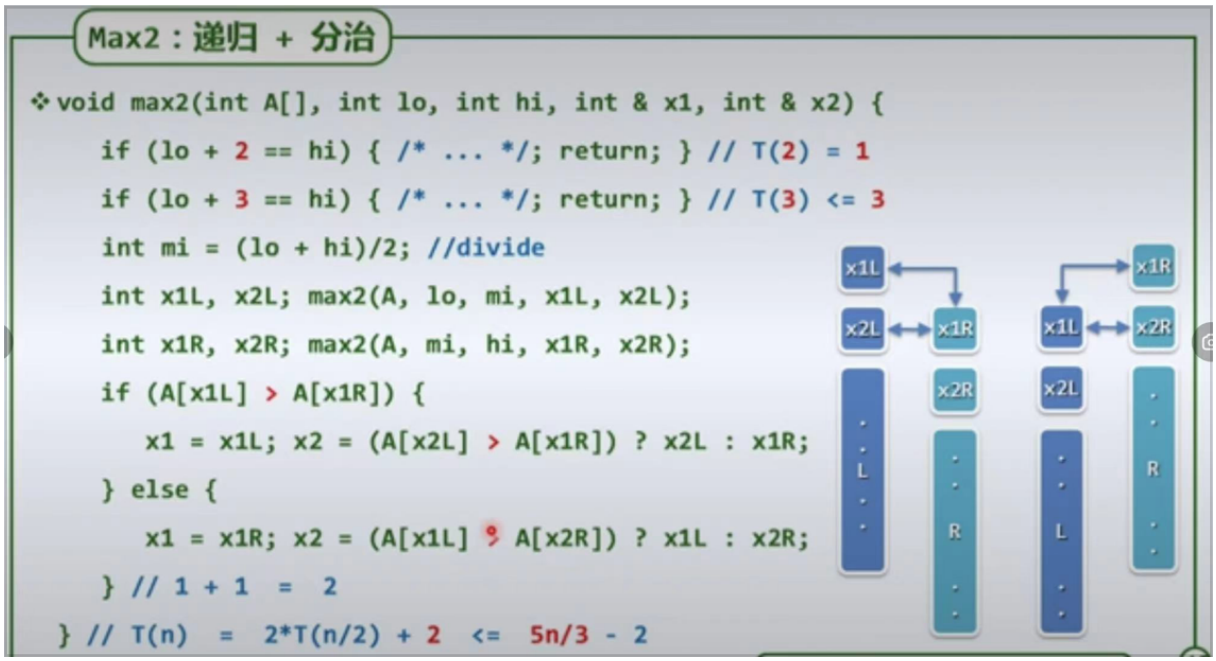
MAX2 : 迭代1

```
从数组区间A[lo,hi)  中找出最大的两个整数A[x1]和a[x2]
元素的比较次数要求最少

void max2(int A[],int lo,int hi,int& x1,int& x2){
// 找到最大值 n-1次
    x1 = lo;
    for (int i = lo+1; i < hi; ++i) {
        if(A[x1] < A[i]){
            x1 = i;
        }
    }
// 找到次大值 n-2 次
    x2 = lo;
    for (int i = lo+1; i < x1; ++i) {
        if (A[x2] < A[i]){
            x2 = i;
        }
    }
    for (int i = x1+1; i < hi; ++i) {
        if (A[x2] < A[i]){
            x2 = i;
        }
    }
}
总数为 θ(2n-3)
```

MAX2: 迭代2

```
void max2(int A[],int lo,int hi,int& x1,int& x2){
    if (A[x1 = lo] < A[x2 = lo +1]){
        swap(x1,x2);
    }
    for (int i = lo + 2; i < hi; ++i) {
        if (A[x2] < A[i]){
            if (A[x1] < A[x2 = i]){
                swap(x1,x2);
            }
        }
    }
}

最好情况:  T(n) = 1 + (n-2)*1 = n-1 = O(1)
最坏情况:  T(n) = 1 + (n-2)*2 = 2*n-3 // 这个与刚才一样
```

MAX2:递归 + 分治




```
void max2(int A[],int lo,int hi,int& x1,int& x2){
    // 两种递归基
    if (lo + 2 == hi){ // T(2) = 1
        /* ... */
        return;
```

```
    }
    if (lo + 3 == hi){ // T(3) <= 3
        /* ... */
        return;
    }

    int mi = (lo + hi) / 2;
    int x1L,x2L;
    max2(A,lo,mi,x1L,x2L);
    int x1R,x2R;
    max2(A,mi,hi,x2L,x2R);
    if (A[x1L] > A[x1R]){
        x1 = x1L;
        x2 = (A[x2L] > A[x1R]) ? x2L : x1R;
    }
    else {
        x1 = x1R;
        x2 = (A[x1L] > A[x2R]) ? x1L : x2R;
    } // 1 + 1 = 2
} // T(n) = 2*T(n/2) + 2 <= 5*n/3 - 2
```