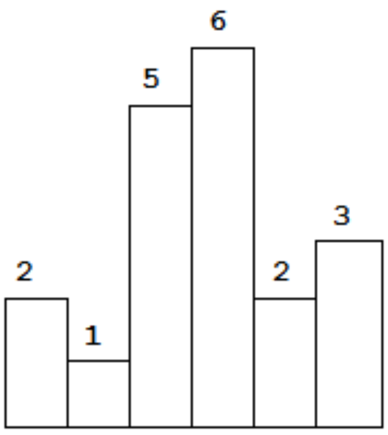
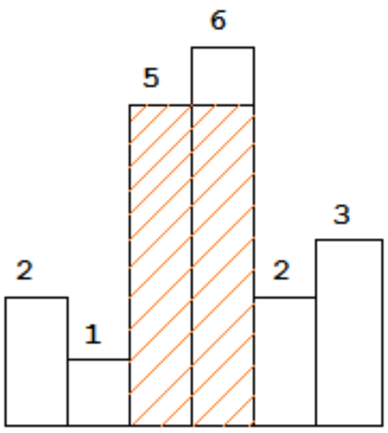
84. Largest Rectangle in Histogram

题目描述：https://leetcode.com/problems/largest-rectangle-in-histogram/

> 给定一个数组，代表每个bar的高度，求可以形成的最大矩形面积。 例如：
>
> 
>
>
> Above is a histogram where width of each bar is 1, given height = [2,1,5,6,2,3].
>
> 
>
>
> The largest rectangle is shown in the shaded area, which has area = 10 unit.
>
> ```
> Given heights = [2,1,5,6,2,3],
> return 10.
> ```

解题思路：

1. 分治算法。求midimum左边右边和整个的最大值。
2. 用栈存储按照增长模式的bar。

代码1：

```
class Solution {
public:
    int largestRectangleArea(vector<int>& heights) {
        int len = heights.size();
        if(len == 0) return 0;
        return calculate(heights, 0, len-1);
    }
    int calculate(vector<int> &heights, int l, int r) {
        if(r - l == 0) return heights[l];
        int mid = l+(r-l)/2;
        int i = mid, j = mid+1;
        int h = INT_MAX, area = 0;
        while(i >= l && j <= r) {
            h = min(h, min(heights[i], heights[j]));
            area = max(area, h*(j-i+1));
            if(i == l) j++;
            else if(j == r) i--;
            else {
                if(heights[i-1] > heights[j+1]) i--;
                else j++;
            }
        }
        cout << "h:" << h << " area:" << area<<endl;
        int ll = calculate(heights, l, mid);
        int rr = calculate(heights, mid + 1, r);
        // cout << "l:" << ll << " r:" << r << " area:" << area << endl;
        return max(max(ll, rr), area);
    }
};
```

代码2:

```
class Solution {
public:
    int largestRectangleArea(vector<int>& heights) {
        stack<int> st;
        heights.push_back(-1);
        int len = heights.size(), res = 0;
        for(int i = 0; i < len; i++) {
            int h = 0, w = 0;
            while(!st.empty() && heights[i] <= heights[st.top()]) {
                h = heights[st.top()];
                st.pop();
                w = i - (st.empty() ? -1 : st.top()) - 1;
                res = max(res, w*h);
            }
            st.push(i);
        }
        return res;
    }
};
```