

Find the nearest smaller numbers on left side in an array

给定一个数组arr，返回每个元素arr[i]其左边所有元素中最小的且位置离arr[i]最近的元素（如果没有则返回-1）组成的集合

1. 初始化 空栈s
2. 从前往后遍历arr[i]，对于每个arr[i]：
 1. 若当前栈不为空且栈顶元素大于等于arr[i]，则循环弹出栈顶元素，直到栈顶元素小于arr[i]或者栈空
 2. 若当前栈为空，代表该元素左边不存在小于该元素的值，输出-1
 3. 若当前栈不为空，输出栈顶元素，即为离arr[i]最近且小于arr[i]的元素
 4. 将当前元素arr[i]入栈

```
#include <bits/stdc++.h>

using namespace std;

typedef vector<int> vi;

void nearestSmaller(vi &arr, int &n) {
    stack<int> s;
    for (int i = 0; i < n; ++i) {
        while (!s.empty() && arr[i] <= s.top()) s.pop();
        if (s.empty()) printf("-1 ");
        else printf("%d ", s.top());
        s.push(arr[i]);
    }
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int n;
        scanf("%d", &n);
        vi arr(n);
        for (int i = 0; i < n; ++i) scanf("%d", &arr[i]);
        nearestSmaller(arr, n);
    }
}
```