

Longest Increasing Subsequence

| 最长递增子序列

令 $dp[i]$ 表示 $arr[0 \sim i]$ 的最长递增子序列长度

考虑第 i 个元素 $arr[i]$ ，有两种情况

1. 对于 $0 < j < i$ ，如果 $arr[j] < arr[i]$ 且 $dp[j] + 1 > dp[i]$ ，则令 $dp[i] = dp[j] + 1$
2. 对于 $0 < j < i$ ，不存在 $arr[j] > arr[i]$ ，则 $dp[i] = 1$

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

using namespace std;

int LIS(vector<int> &arr, int &N) {
    vector<int> dp(N);
    dp[0] = 1;
    int maxLen = dp[0];
    for (int i = 1; i < N; ++i) {
        dp[i] = 1;
        for (int j = 0; j < i; ++j) {
            if (arr[i] > arr[j] && dp[i] < dp[j] + 1) {
                dp[i] = dp[j] + 1;
                maxLen = max(maxLen, dp[i]);
            }
        }
    }
    return maxLen;
}

int main() {
    int T;
    scanf("%d", &T);
    while (T--) {
        int N, num;
        vector<int> arr;
        scanf("%d", &N);
        for (int i = 0; i < N; ++i) {
            scanf("%d", &num);
            arr.push_back(num);
        }
        printf("%d\n", LIS(arr, N));
    }
    return 0;
}
```