Maximum sum increasing subsequence

LIS （最长递增子序列）

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

sumOfLIS

只需要将dp从表示长度改为表示和即可

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

using namespace std;


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


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", sumOfLIS(arr, N));
    }
    return 0;
}
```