Maximum length subsequence with difference between adjacent elements as either 0 or 1

> 最长子序列——需要满足子序列中相邻元素之差不是0就是1

思路类似于LIS，只是在更新DP时的判断条件不同

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

using namespace std;

int LIS(vector<int> &arr, int &N) {
    vector<int> dp(N, 1);
    int maxLen = dp[0];
    for (int i = 1; i < N; ++i) {
        for (int j = 0; j < i; ++j) {
            if ((abs(arr[i] - arr[j]) == 0 || abs(arr[i] - arr[j]) == 1) && 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;
        scanf("%d", &N);
        vector<int> arr;
        for (int i = 0; i < N; ++i) {
            scanf("%d", &num);
            arr.push_back(num);
        }
        printf("%d\n", LIS(arr, N));
    }
    return 0;
}
```