**★★★★☆**

**題組：Contest Volumes**

**題號：11456: Trainsorting**

**解題者：朱怡宣**

**解題日期：2025年5月1日**

**題意：**

給定T組測資，每組先輸入一個正整數n（1 ≤ n ≤ 2000），代表車廂數量；接著讀入n個正整數，表示這些車廂依序到達時的重量。當一輛車廂到達時，可選擇將它接到列車的前端、或後端，也可選擇丟棄。最終接上的車廂，必須從車頭到車尾呈現嚴格遞減（每輛後接車廂的重量必須比前一輛小）。求在符合以上條件下，最多可接上多少輛車。

**題意範例：**

1

3

1 🡺 3

2

**解法：**

1. 反轉車廂重量序列

將原始到達順序的車廂重量存入陣列，然後反轉。如此一來，只要從左至右依序掃描，就能同時模擬「放到尾端」與「放到前端」兩種插入方式。

1. 計算最長遞增和遞減子序列

定義兩個陣列：

* 最長遞增長度LIS[i]（對應將該車放在列車尾端）
* 最長遞減長度LDS[i]（對應將該車放在列車前端）

從頭遍歷反轉後的矩陣，若w[j]<w[i]，更新LIS。反之則更新LDS。

1. 組合bitonic序列並計算出最多可接上幾輛車廂。

以每個 i 為轉折點，可組成長度 = LIS[i] + LDS[i] – 1。

**解法範例：**

無

**討論：**

1. 反轉序列的必要性：若不反轉，原本的『前端插入』就無法在一次從左到右的掃描裡，用『尾部延伸』來模擬，導致 LDS 無法涵蓋所有前端插入的可能性。
2. Bitonic：指的是先嚴格遞增，到某個轉折點後，再嚴格遞減的數列。在本題中，把遞增（LIS）與遞減（LDS）合併，就是在找最大的bitonic子序列長度。

**程式：**

#include <stdio.h>

#include <stdlib.h>

#define MAXN 2000

int main() {

    int T;

    if (scanf("%d", &T) != 1) return 0;

    while (T--) {

        int n;

        scanf("%d", &n);

        if (n == 0) {

            printf("0\n");

            continue;

        }

        static int w[MAXN];

        for (int i = 0; i < n; i++) {

            scanf("%d", &w[i]);

        }

        // 反轉序列：arr[0] 對應最後到達的車，方便做 bitonic subsequence

        static int arr[MAXN];

        for (int i = 0; i < n; i++) {

            arr[i] = w[n - 1 - i];

        }

        static int LIS[MAXN], LDS[MAXN];

        int best = 0;

        // 計算 LIS 與 LDS，並同時計算最大 bitonic 長度

        for (int i = 0; i < n; i++) {

            LIS[i] = LDS[i] = 1;

            for (int j = 0; j < i; j++) {

                if (arr[j] < arr[i] && LIS[j] + 1 > LIS[i]) {

                    LIS[i] = LIS[j] + 1;

                }

                if (arr[j] > arr[i] && LDS[j] + 1 > LDS[i]) {

                    LDS[i] = LDS[j] + 1;

                }

            }

            // 以 i 為「樞紐」，列車一側由重量較小的車（遞增），另一側由重量較大的車（遞減）

            int len = LIS[i] + LDS[i] - 1;

            if (len > best) best = len;

        }

        printf("%d\n", best);

    }

    return 0;

}