CS 577: Introduction to Algorithms	10/29/2015
Homework 7	
Instructor: Dieter van Melkebeek	TA: Kevin Kowalski, Andrew Morgan, Bryce Sandlund

This is a second assignment on dynamic programming. Good luck!

Review problems

1. Give an efficient algorithm that takes as input two binary sequences a and b, and outputs the smallest length of a sequence c such that both a and b are subsequences of c.

   For example, if $a = 011$ and $b = 0101$ then the answer is 4, as witnessed by $c = 0101$.

2. Consider the multiplication defined in Table 1. For example, $ab = b$ and $ba = c$. (Note that this multiplication is neither associative nor commutative.)

   Table 1: Multiplication table

	a	b	c
a	b	b	a
b	c	b	a
c	a	c	c

   You are given an element of $\{a, b, c\}^+$, which is a nonempty string of symbols a, b, and c. Give an efficient algorithm that decides whether or not it is possible to parenthesize the string in such a way that the value of the resulting expression is a. For example, on input $bbbbac$ your algorithm should return *yes* because $((b(bb))(ba))c = a$.

Graded written problem

3. [10 points] In modern origami (the Japanese art of paper folding), one typically starts with a square sheet of paper and attempts to transform this square into a three-dimensional animal, geometric object, or any other sculpture one can think of using nothing but a sequence of folds. In traditional 17th-18th century origami, however, the starting shape of the paper was less strictly prescribed.

   Hiro has stumbled across a book containing instructions for n origami sculptures from this early period, each of which starts from rectangular paper of size $a_i \times b_i$ where a_i and b_i are positive integers. He would like to make a diorama containing as many of these (not necessarily distinct) sculptures as possible, but he only has access to a single sheet of paper of size $A \times B$ (where A and B are also positive integers) and no scissors. By folding the paper carefully and tearing along the crease, Hiro is confident that he can make perfect horizontal and vertical cuts across an entire sheet of paper, splitting the sheet into two. However, the two new edges created by each of these cuts are *frayed* and look worse than the original edges of the $A \times B$ sheet, so Hiro would like to minimize the number of these edges as well.

Give a polynomial-time algorithm that on input $A, B, a_1, \ldots, a_n, b_1, \ldots, b_n$, computes the maximum number of sculptures that can be made from the starting sheet of paper, as well as the minimum number of frayed edges that must be visible when Hiro creates this maximum number of sculptures.

Additional written problem

4. Consider the setting of the weighted interval scheduling problem from class, but with two identical machines rather than a single machine. In order to execute a job, one of the machines needs to be reserved for the entire duration of the job.

   Develop a polynomial-time algorithm to construct an optimal schedule.

Optional programming problem

5. [2.5 points] Solve SPOJ problem Square Brackets (problem code SQRBR).

Challenge problem

6. Problem "Blocks" from the ACM-ICPC (see attachment). Your algorithm should run in time polynomial in n.

Problem A
Blocks

Input: Standard Input
Output: Standard Output
Time Limit: 10 Seconds

Some of you may have played a game called 'Blocks'. There are **n** blocks in a row, each box has a color. Here is an example: Gold, Silver, Silver, Silver, Silver, Bronze, Bronze, Bronze, Gold.

The corresponding picture will be as shown below:

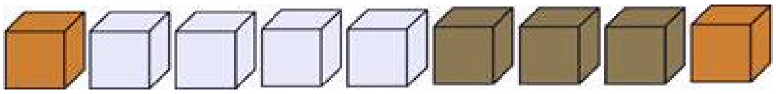

Figure 1

If some adjacent boxes are all of the same color, and both the box to its left(if it exists) and its right(if it exists) are of some other color, we call it a 'box segment'. There are **4** box segments. That is: gold, silver, bronze, gold. There are **1**, **4**, **3**, **1** box(es) in the segments respectively.

Every time, you can click a box, then the whole segment containing that box DISAPPEARS. If that segment is composed of **k** boxes, you will get **k*****k** points. for example, if you click on a silver box, the silver segment disappears, you got **4*****4**=**16** points.

Now let's look at the picture below:

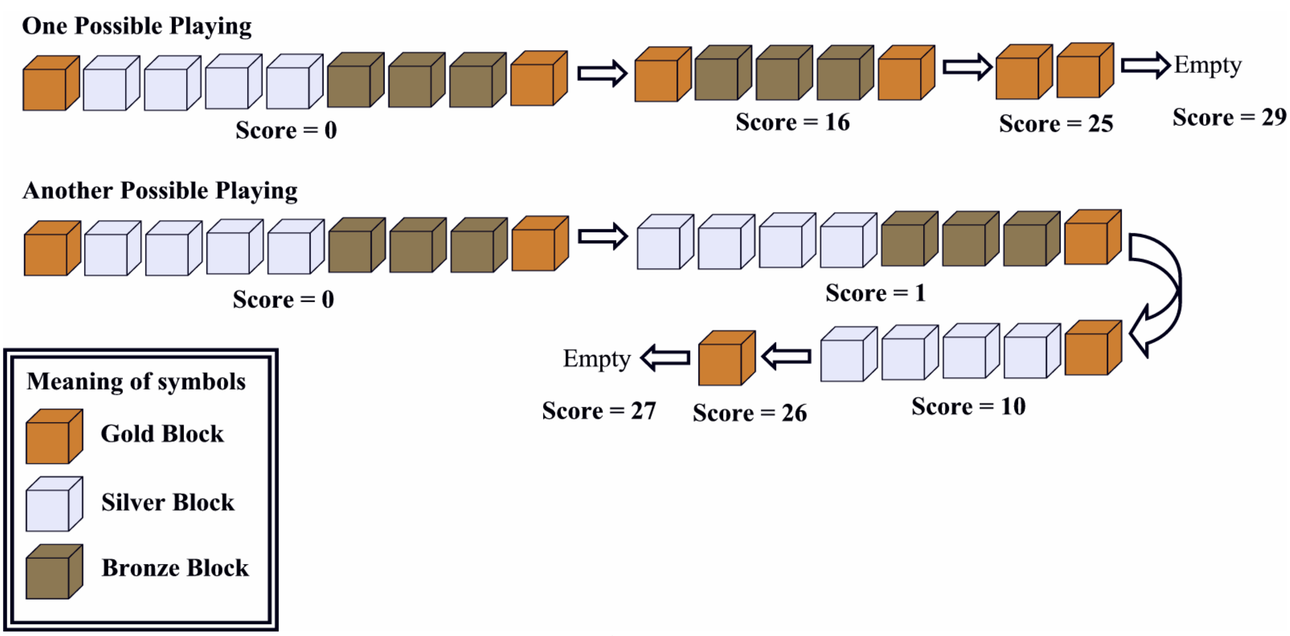

Figure 2

The first one is OPTIMAL.

Find the highest score you can get, given an initial state of this game.

Input

The first line contains the number of tests **t(1<=t<=15)**. Each case contains two lines. The first line contains an integer **n(1<=n<=200)**, the number of boxes. The second line contains **n** integers, representing the colors of each box. The integers are in the range **1~n**.

Output

For each test case, print the case number and the highest possible score.

Sample Input

```
2
9
1 2 2 2 2 3 3 3 1
1
1
```

Output for Sample Input

```
Case 1: 29
Case 2: 1
```

Problemsetter: Rujia Liu, Member of Elite Problemsetters' Panel
Special Thanks: Cailiang Liu & Rongjing Xiang from IOI2003 China National Training Team