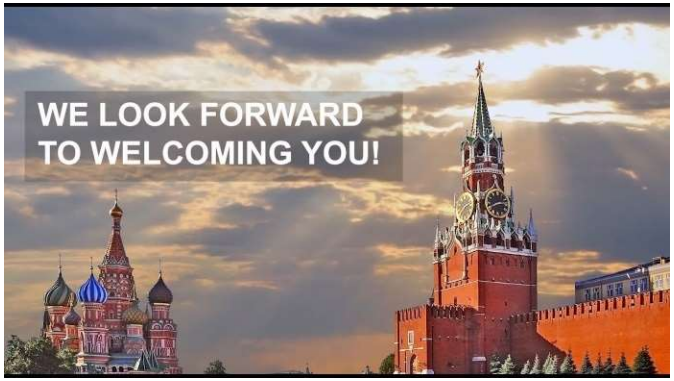
This course material is now made available for public usage.
Special acknowledgement to School of Computing, National University of Singapore
for allowing Steven to prepare and distribute these teaching materials.

CS3233
Competitive Programming

Dr. Steven Halim

Let's Talk CP

icpc international collegiate programming contest
icpc.foundation



Singapore 2020

32ND INTERNATIONAL
OLYMPIAD IN INFORMATICS
SINGAPORE

Menu (1st session : 6.10-7.30pm)

(0). Self (Re-)Introduction ☺

1. Benefits of Competitive Programming
   - Hopefully inspiring for Malaysian (young) programmers ☺
2. Competitive Programming
   - Introduction
   - Example
   - Tips to be Competitive

2nd session (7.40-9.50pm): Graph/Tree Algorithms

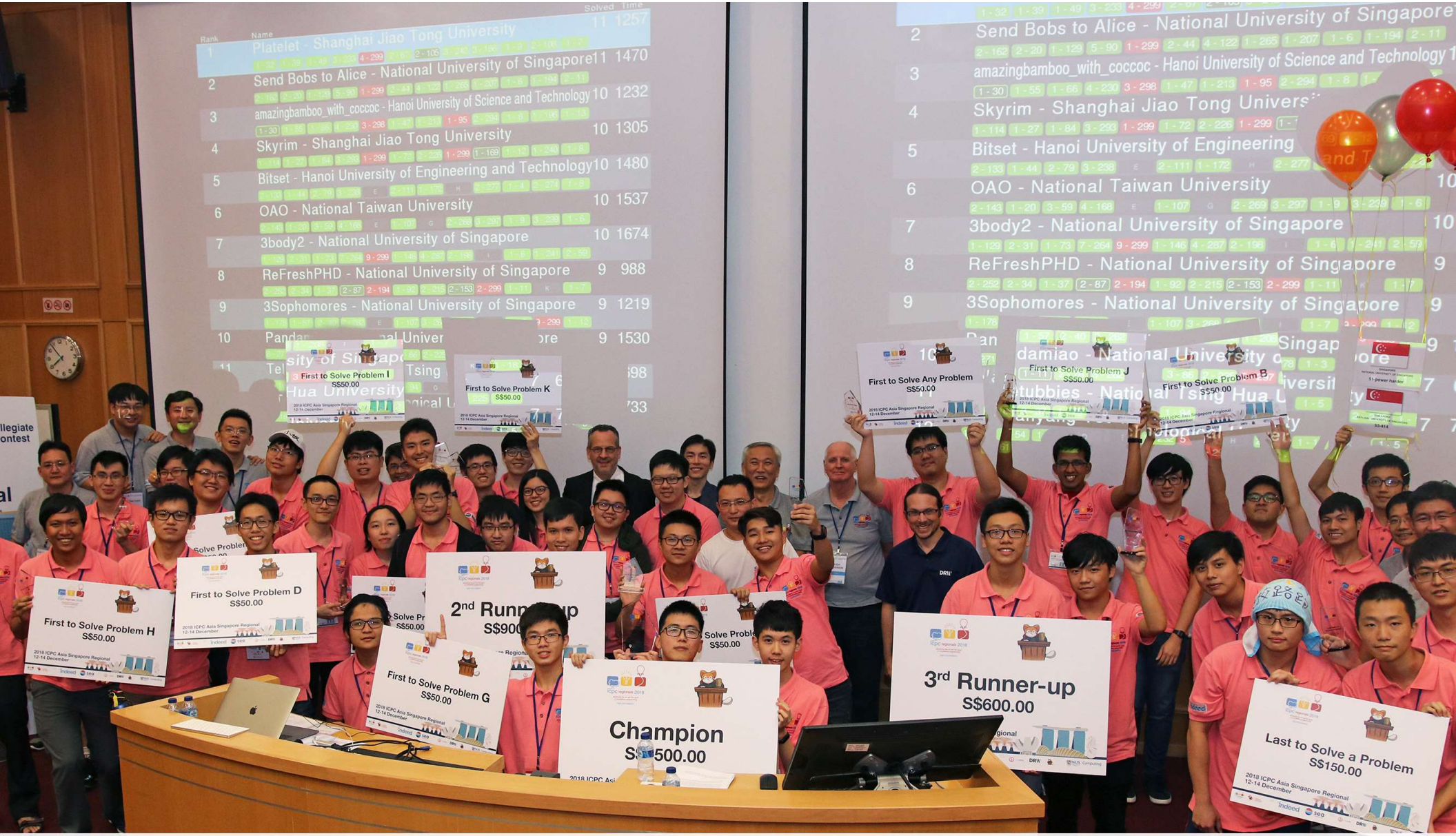
~~3rd session (8.30-9.30pm): Dynamic Programming (deferred to later)~~

COVID-19 and NUS latest policy

- https://emergency.nus.edu.sg/ says:
  "No physical class bigger than 50, has to switch to e-Learning"
  - I heard 80+ registered participants
    - Actually peaked around 70+, averaging 50+
- If I happen to get the virus,
  I will be isolated for until I am well (historically 14++ days)
  - That is a risk that is too dangerous to take considering IOI 2020 Winter Meeting that will take place on 24-28 February 2020 :O and my role there (IOI 2020 Deputy Director)

Recent CS3233 @ Singapore
Jan-Apr 2019

ICPC Asia Singapore 2015



ICPC Asia Singapore 2018

National University of Singapore (NUS) teams in ICPC (Regionals) (2013-4-5)

NUS teams (usually) do (very) well in recent Asia regionals, winning 5x in Jakarta, 2x in Manila, 1x in Nakhon Pathom, 1x in Yangon, **1x in KL**, and often in top 3 in the past 11 years

National University of Singapore (NUS) teams in ICPC (Regionals) (2017)

NUS teams (usually) do (very) well in recent Asia regionals, winning 5x in Jakarta, 2x in Manila, 1x in Nakhon Pathom, 1x in Yangon, **1x in KL**, and often in top 3 in the past 11 years

National University of Singapore (NUS) teams in ICPC (Regionals) (2018)

NUS teams (usually) do (very) well in recent Asia regionals, winning 5x in Jakarta, 2x in Manila, 1x in Nakhon Pathom, 1x in Yangon, **1x in KL**, and often in top 3 in the past 11 years

National University of Singapore (NUS) teams in ICPC (Regionals) (2019)

NUS teams (usually) do (very) well in recent Asia regionals, winning 5x in Jakarta, 2x in Manila, 1x in Nakhon Pathom, 1x in Yangon, **1x in KL**, and often in top 3 in the past 11 years

If you are interested to read: https://algorithmics.comp.nus.edu.sg/2019-KualaLumpur-Report-team-3body3.pdf

Scoreboard - ICPC 2019 Kuala Lu

Not secure | iium.edu.my/acmicpc/2019/regional/scoreboard.html

DOMjudge

contest over

RANK	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J
1	3body3 National University of Singapore	10 1411	265 1 try	211 3 tries	200 1 try	23 1 try	28 2 tries	207 5 tries	50 1 try	100 1 try	147 1 try	40 1 try
2	UKUNICHIA University of Aizu	8 854	275 1 try	143 1 try		58 1 try	21 1 try	5 tries	28 1 try	123 2 tries	81 1 try	105 1 try
3	Quantasaurus Tiguerra Universitas Indonesia	8 1239	1 try	181 1 try		141 2 tries	11 1 try	284 2 tries	29 1 try	270 2 tries	144 1 try	99 2 tries
4	NTU_stdio National Taiwan University	8 1270	292 3 tries	261 2 tries		73 1 try	14 1 try		41 1 try	116 3 tries	245 2 tries	108 1 try
5	NP-TRAN Hanoi University of Science and Technology	7 760	3 tries	78 2 tries		37 2 tries	45 1 try		60 1 try	142 3 tries	123 2 tries	175 1 try
6	Pokemon Goh National University of Singapore	7 810		163 1 try		59 3 tries	10 1 try	7 tries	68 1 try	121 3 tries	198 1 try	91 2 tries
						76 ies	16 3 tries	264 2 tries	21 1 try	51 2 tries		87 1 try
						1 ry	20 1 try		30 1 try	1 try	170 2 tries	93 1 try
						4 ies	12 1 try		19 1 try	53 2 tries		145 2 tries
						6 ies	18 1 try	185 7 tries	79 1 try	10 tries		106 1 try
						8 ies	18 1 try	6 tries	70 2 tries	292 2 tries		182 1 try
						70 ry	17 1 try	122 6 tries	36 1 try	14 tries		81 1 try
						21 ies	28 1 try		160 1 try			163 1 try
						79 ies	22 1 try		54 2 tries	102 3 tries		294 5 tries
						7 ry	27 1 try		114 1 try			197 2 tries
						6 ies	15 1 try		43 1 try	2 tries		200 3 tries
							36 1 try		101 1 try	4 tries		73 2 tries

SoC Teams Performance History (1)

Recent ICPC Regional Contests

- 2004: **1st in Kanpur**, but no top 3 at all in 2005-2007
- 2008: 3rd in Kanpur
- 2009: 2nd in Phuket; 3rd in Manila
- 2010: 6th* in Kuala Lumpur
- 2011: 7th* in Phuket; 5th* in Kuala Lumpur
- 2012: 3rd in Jakarta
- 2013: **1st in Jakarta**
- 2014: **1st in Jakarta**, 2nd in Bangkok
- 2015: **1st in Jakarta**, 3rd in Singapore
- 2016: 2nd* in Jakarta, 3rd in Yangon
- 2017: **1st in Jakarta**, **1st in Manila**, 2nd in NakhonPathom
- 2018: **1st in NakhonPathom**, **1st in Yangon**, 2nd in Singapore, 3rd in Jakarta
- 2019: **1st in Jakarta**, **1st in KualaLumpur**, **1st in Manila**, 3rd in Taipei-Hsinchu

More history in:
http://algorithmics.comp.nus.edu.sg

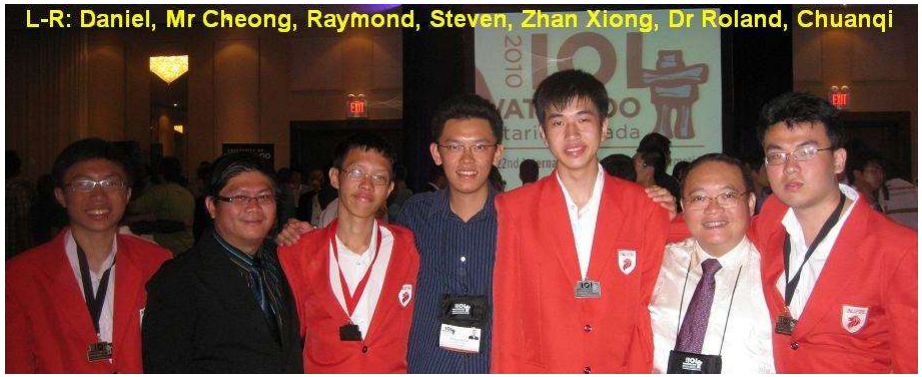
SoC Teams Performance History (2)

Recent ICPC World Finals

- Barren years: 2002, 2004, **2006-2007-2008 (no CS3233)** ☹, 2011 (the only blot in 2010s decade)
- 1999: Joint-18
- 2000: Joint-22
- 2001: Joint-29
- 2003: Joint-13
- 2005/Shanghai, China: Melvin, Junbin, Yunsong: Hon. Mention
- 2009/Stockholm, Sweden: Duc, Tien, Phong: Hon. Mention
- 2010/Harbin, China: Duc, Tien, Phong: Hon. Mention
- 2012/Warsaw, Poland: Zi Chun, Harta, Phuong: Hon. Mention
- 2013/St Petersburg, Russia: Harta, Phuong, Sy Nguyen, Joint-48/120
- 2014/Ekaterinburg, Russia: Jonathan, Nathan, Quang, Joint-19/125
- 2015/Marrakesh, Morocco: Jonathan, Nathan, DatVu, Joint-28/128
- 2016/Phuket, Thailand: Thanh Trung, Sy Nguyen, Hung Tam, Joint-14/128 (current best)
- 2017/Rapid City, USA: Muh Rais FM, Agus SH*, How Si Wei*, Joint-20/138
- 2018/Beijing, China: Phan Duc Nhat Minh*, DatVu, Manh, Joint-56/140
- 2019/Porto, Portugal: Wei Heng*, Bernard Teo*, Sidhant Bansal*, Joint-62/135
- 2020/Moscow, Russia: Wei Heng*, Bernard Teo*, Sidhant Bansal*, TBA?

Singapore teams in IOI (1)

Team photos from the past 10 years: 2010-2013



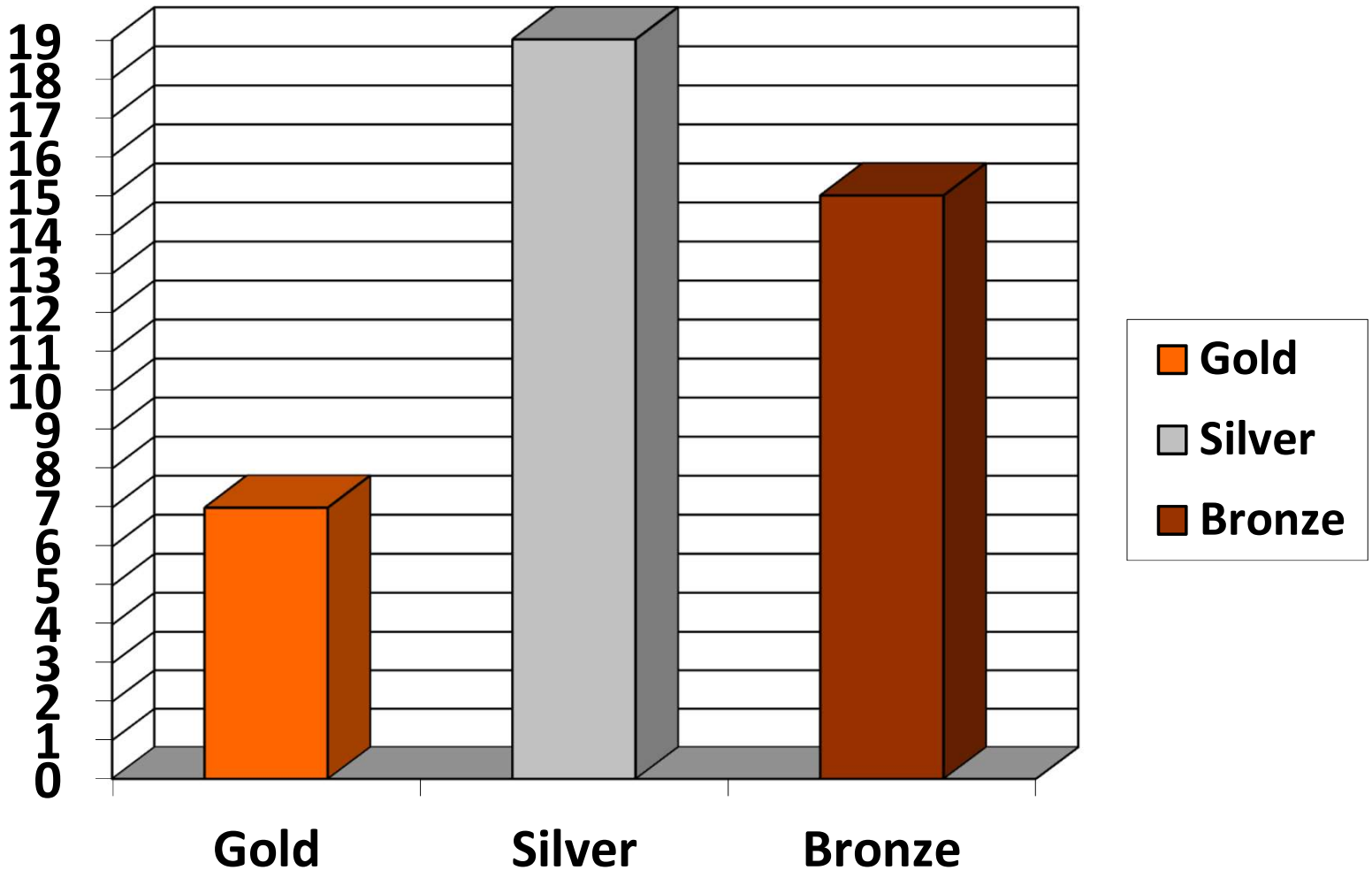
Singapore teams in IOI (2)

Team photos from the past 10 years: 2014-2017

Singapore teams in IOI (3)

Team photos from the past 10 years: 2018-2019

SG IOI Team Medal Tally 2009-19



CS3233 (Top) Alumni

facebook

Google

Microsoft

indeed

jumptrading

sea
connecting the dots

HRT

DRW

商汤
sensetime

Grab

The Lower Bound of Programming Contests in 2020s

[will be hard to do this via webinar, but try to answer some live questions if you can]

COMPETITIVE PROGRAMMING 4

Competitive Programming

Given <u>well-known</u> Computer Science problems, solve them <u>as fast as possible</u>!

- Not about "software engineering"
- Solve judge's test data correctly
- Run fast enough
- Well-known = not research problems!
- It is a game (academic sport)
- Problems in our target contests (IOI & ICPC) have this characteristic!
- **<u>Not</u>** the end goal!

Demo (UVa 10684 – The jackpot)

This demo illustrates contestant's type:

A. The blurry one
B. Give up
C. Slow
D. Competitive programmer
E. Very competitive programmer

Problem D - The jackpot

Time Limit: 1 second

Memory Limit: 1 Mb

Background

As Manuel wants to get rich fast and without too much work, he decided to make a career in gambling. Initially, he plans to study the gains and losses of players, so that, he can identify patterns of consecutive wins and elaborate a win-win strategy. But Manuel, as smart as he thinks he is, does not know how to program computers. So he hired you to write programs that will assist him in elaborating his strategy.

The Problem

Your first task is to write a program that identifies the maximum possible gain out of a sequence of bets. A bet is an amount of money and is either winning (and this is recorded as a positive value), or losing (and this is recorded as a negative value).

Input

The input set consists of a positive number $\mathbf{N} \leq \mathbf{10000}$, that gives the length of the sequence, followed by **N** integers. Each bet is an integer greater than **0** and less than **1000**.

The input is terminated with $\mathbf{N} = \mathbf{0}$.

Output

For each given input set, the output will echo a line with the corresponding solution. If the sequence shows no possibility to win money, then the output is the message "Losing streak."

Sample input

```
5
12 -4
-10 4
9
3
-2 -1 -2
0
```

Sample output

```
The maximum winning streak is 13.
Losing streak.
```

David Deharbe 2004-06-19

Anatomy of a Problem

- Background story/problem description
  - Can be important but can also be deceiving…
- Input and Output description
  - Usually written in formal manner
  - Older = *multiple test cases* per input file
  - Newer = *one* test case per input file
- Sample Input and Sample Output
  - Usually very trivial, you need to come up with stronger/trickier test cases by yourself
- Hints or Footnotes
- [NOI/IOI version: There are usually subtasks with lower scores than the full solution that can be solved with easier algorithm]

Problem D - The jackpot

Time Limit: 1 second

Memory Limit: 1 Mb

Background

As Manuel wants to get rich fast and without too much work, he decided to make a career in gambling. Initially, he plans to study the gains and losses of players, so that, he can identify patterns of consecutive wins and elaborate a win-win strategy. But Manuel, as smart as he thinks he is, does not know how to program computers. So he hired you to write programs that will assist him in elaborating his strategy.

The Problem

Your first task is to write a program that identifies the maximum possible gain out of a sequence of bets. A bet is an amount of money, a number of units, and this can be a gain or a loss (and this is recorded as a negative value).

Input

The input set consists of a positive number $N \leq 10000$, that gives the length of the sequence, followed by **N** integers. Each bet is an integer greater than **0** and less than **1000**.

The input is terminated with $N = 0$.

Output

For each given input set, the output will echo a line with the corresponding solution. If the sequence shows no possibility to win money, then the output is the message "Losing streak."

Sample input

```
5
12 -4
-10 4
9
3
-2 -1 -2
0
```

Sample output

```
The maximum winning streak is 13.
Losing streak.
```

David Deharbe 2004-06-19

- Background story
  - Totally irrelevant here (only to give context)
- Problem description
  - The important one
- Input and Output description
  - Here, multiple test case format
  - (but actually # of TC is unclear here :O)
- Sample Input and Sample Output
  - Usually very trivial
  - Here, only small cases are shown
- Hints or Footnotes [none here]

Demo (UVa 10684 – The jackpot)

This demo illustrates contestant's type:

A. The blurry one

B. Give up

index	0	1	2	3	4
A	12	-4	-10	4	9
Start from 0/prefix sum	12	8	-2	2	11
Start from 1		-4	-14	-10	-1
Start from 2			-10	-6	3
Start from 3				4	13
Start from 4					9

Demo (UVa 10684 – The jackpot)

This demo illustrates contestant's type:

A. The blurry one

B. Give up

C. Slow

D. Competitive programmer

E. Very competitive programmer

It has a name:
Kadane's algorithm

index	0	1	2	3	4
lastA	12	-4	-10	4	9
prefix sum	12	8	-2	2	11
running sum	12	8	-2 → 0	4	13
ans	12	12	12	12	13

Demo (UVa 10684 – The jackpot)

```
#include <bits/stdc++.h> // not C++ standard, but OK for CP
using namespace std;
int main() {
  int N;
  while (scanf("%d", &N), N) {
    int sum = 0, ans = 0;
    for (int i = 0; i < N; ++i) {
      int lastA; scanf("%d", &lastA); // on the fly
      sum += lastA;
      ans = max(ans, sum);
      if (sum < 0) sum = 0; // Max 1D Range Sum, Kadane's
    }
    if (ans > 0)
      printf("The maximum winning streak is %d.\n", ans);
    else
      printf("Losing streak.\n");
  }
  return 0;
}
```

CP4, Chapter 1

TIPS TO BE COMPETITIVE

Tip 1: Type Fast & Correct

No kidding, this can be important!

Try this…

- https://www.typingtest.com, random 1 minute test
  - Steven's: ~85-95 wpm
  - Felix's: ~55-65 wpm
  - Suhendry's: ~70-80 wpm

Familiarize yourself with the positions of the following keyboard keys:

- (,),{,},[,],<,>,',",&,|,!,etc

Tip 2: Identify Problem Types

1. Ad Hoc
2. Data Structure
3. Complete Search
4. Divide and Conquer
5. Greedy
6. Dynamic Programming
7. Graph
8. Mathematics
9. String Processing
10. Comp. Geometry
11. Rare/Harder Ones

Let's see VisuAlgo

Better Classification

A. I have solved this type before
   - I am sure that I can re-solve it again (and fast; *not too long*)
   - This is the preferred classification, **maximize this**
B. I have seen this type before
   - But that time I know I cannot solve it yet
   - You have to **minimize this**
C. I have not seen this type before
   - I may (*or may not*) be able to solve it now
   - To *win* a contest, you need to frequently able solve this type especially if the solution is derived from 'basic principles'

Tip 3: Do Algorithm Analysis

This is taught in more details in a typical "Design and Analysis of Algorithm" course in University Computer Science curriculum

In Competitive Programming, we will just learn the *basics* required for dealing with IOI/ICPC problems

- See the constraints in the problem statement
- Conjure the simplest algorithm that works!
- Do some basic analysis to convince that it will work *before* we start coding…

Our "Reference" Table

n	Worst AC Algorithm	Comment
≤ [10..11]	$O(n!)$, $O(n^6)$	e.g. Enumerating a permutation
≤ [15..18]	$O(2^n * n^2)$	e.g. DP TSP
≤ [18..22]	$O(2^n * n)$	e.g. DP with Bitmask
≤ [24..25]	$O(2^n)$	e.g. try all subsets of n items
≤ 100	$O(n^4)$	e.g. DP with 3 dimensions + O(n) loop
≤ 450	$O(n^3)$	e.g. Floyd Warshall's
≤ 1.5K	$O(n^{2.5})$	e.g. Hopcroft Karp's
≤ 2.5K	$O(n^2 \log_2 n)$	e.g. Two nested loops with a tree-related DS
≤ 10K	$O(n^2)$	e.g. Bubble/Selection/Insertion sort
≤ 200K	$O(n\sqrt{n})$	e.g. square root decomposition technique
≤ 4.5M	$O(n \log_2 n)$	e.g. Merge Sort
≤ 100M	O(n) and sub O(n)	Usually, contest problems has n ≤ 1M

Red: New CP4 bound compared to CP3

Quick Test – Identification (1)

Given a multiset **S** of **M** = **100K** integers, we want to know how many different integers that we can form if we pick two integers from **S** and sum them

The multiset **S** contains prime numbers ≤ 20K

What is your chosen algorithm and/or data structure?

Answer: Data compression first, there are only **N** = 2 262 distinct prime numbers under 20K in set **S'** (from multiset **S**, removing duplicates until there are at most 2 per prime). At the end we have at most **N** = 4 524 items. Then we can just do O(**N**^{2}) algorithm, use unordered_set to help us count the number of unique sums

Tip 4: Master Prog Languages

You should master at least one **(preferably more)** programming languages

- Reduce the amount of time looking at references
- Use shortcuts, macros, avoid comments
- Use libraries whenever possible

Idea: Once you figure out a solution for a problem, you are able to translate it into a **bug-free code**, **and do it fast!**

Quick Test – Language (1)

```
int counter1 = 0; // array A and B have been initialized earlier
int counter2 = 0;
for (int i = 0; i < n; ++i) // n is guaranteed to be even
{
  if (A[i] == B[i])
  {
    counter1++;
  }
  else
```

Rewrite this C++ code into a (much) shorter code that does the same thing, assume n is even

```
for (int i = 0, c = 0; i < n; ++i) c += (A[i] == B[i]);
printf("%sbalanced lah\n", (c == n/2 ? "" : "un"));
}
if (counter1 == counter2)
{
  printf("balanced lah\n");
}
else
{
  printf("unbalanced lah\n");
}
```

Quick Test – Language (2)

- I have a 2D integer array called A of size N x M (big)

f A can be positive/0/neg

inate (R, C) inside A, I wa

ect neighbours of (R, C) a

- Here a neighbour is defined as the 8 cells

```
// we use sentinel technique, by storing the input in an
// (N+2)*(M+2) matrix, mapping index (i, j) to (i+1, j+1)
int d[8][2]={{ 1, 1},{ 1, 0},{ 1,-1},{ 0,-1},  // SE,S,SW,W
             {-1,-1},{-1, 0},{-1, 1},{ 0, 1}}; // NW,N,NE,E
int ans = 0;
for (int i = 0; i < 8; ++i) {
  int nr = R+1+d[i][0], nc = C+1+d[i][1]; // +1 offset
  ans += (A[nr][nc]>0);
}
printf("%d\n", ans);
```

Tip 5: The Art of Testing Code

Ultimately, we want "Accepted (AC)" verdict ☺

- i.e., Our code passes the judge's secret test data
- For ICPC: All or Nothing; For IOI/NOI: There may be subtasks

However, we may instead be given: ☹

- Presentation Error (PE)
- Wrong Answer (WA)
- Time Limit Exceeded (TLE)
- Memory Limit Exceeded (MLE)
- Runtime Error (RTE)

Tip 6: Practice…

Relevant Online Judges

- MAIN: Kattis Online Judge
  - https://
- MISC: ~~Uni~~
  - https://
- MISC: ICP
  - https://i

#	USER	SCORE [?]
4	Bjarki Ágúst Guðmundsson	7044.9
5	Josenildo Silva	6994.5
6	Nick Wu	6836.8
7	Doug Goodman	5927.0
8	Matthew Ng	5556.2
9	Steven Halim	5496.4
10	Andreas Björklund	4576.8
11	Bay Wei Heng	4548.6

NUS 3body3

ognito!)

M-ICPC Live Archive

Submissions Statistics

3200

5 | 10 | 50 | 100 | 500 | ALL

Problem		Verdict	Lang	Time	Best	Rank	Submit Time
12049 - Just Prune The List	discuss	Accepted	C++11	0.150	0.020	230	4 days ago
12049 - Just Prune The List	discuss	Accepted	C++11	0.320	0.020	753	4 days ago
12049 - Just Prune The List	discuss	Wrong answer	C++11	-	0.020	-	4 days ago
10125 - Sumsets	discuss	Wrong answer	C++11	-	0.000	-	2017-10-09 14:51
10125 - Sumsets	discuss	Wrong answer	C++11	-	0.000	-	2017-10-09 14:32
10125 - Sumsets	discuss	Wrong answer	C++11	-	0.000	-	2017-10-09 14:29
10125 - Sumsets	discuss	Wrong answer	C++11	-	0.000	-	2017-10-09 14:28

Tip 7: Team Work (ICPC Only)

Not Applicable for NOI/IOI

- Practice coding on a blank paper
- Practice coding by not reading the problem statement at all, but having another team member digest the solution to you
- Submit and print strategy, if your code is not AC, debug the code on that printed paper
- Prepare test data challenges
- Read all currently unsolved problems to double/triple check your team-mate initial ideas
- The X-factor

More details in the next segments:

Some Graph Algorithms (7.40-9.50pm)

~~Some Dynamic Programming (deferred to another time)~~

LET'S START OUR JOURNEY