Programming Contest Training

Hamedan University of Technology

Fall 2023

Instructor: Ali Mohammadpour

ali[DOT]mohammadpour[AT]aut[DOT]ac[DOT]ir

PCT.01. Introduction

Introduction

- What is Programming Contest
- Why Do Programming Contests?
  - You can learn:
    - Many useful algorithms, mathematical insights
    - How to code/debug quickly and accurately
    - How to work in a team
  - Then you can rock in classes, job interviews, etc.
  - It’s also fun! coffee and teamwork and balloons
- Carrier Benefits

Prerequisites

- Introduction to programming experience in true programming language
    - You'll be coding in either C/C++, Java, or Python
- Good mathematical insight
- Most importantly, eagerness to learn
- Practice, Practice, Practice

Topics

- Introduction
- Algorithm Complexity
- Mathematics
- Data structures
- Dynamic programming (DP)
- Combinatorial games
- Graph algorithms
- Shortest distance problems
- Network flow
- Geometric algorithms
- String algorithms

Programming Contests

- ACM-ICPC
  - Pacific Northwest Regional
  - World Finals
- IEEEXtrime
  - 24 hour programming contest
- Online Contests (Worldwide)
  - TopCoder, Codeforces
  - Google Code Jam
- Online Contests (Iranian Community)
  - Sharif Judge (Online, Archived)
  - Quera (Online, Archived)
  - ACM-ICPC (Sharif University of Technology)
  - ACPC (Amirkabir University of Technology)

How to Practice

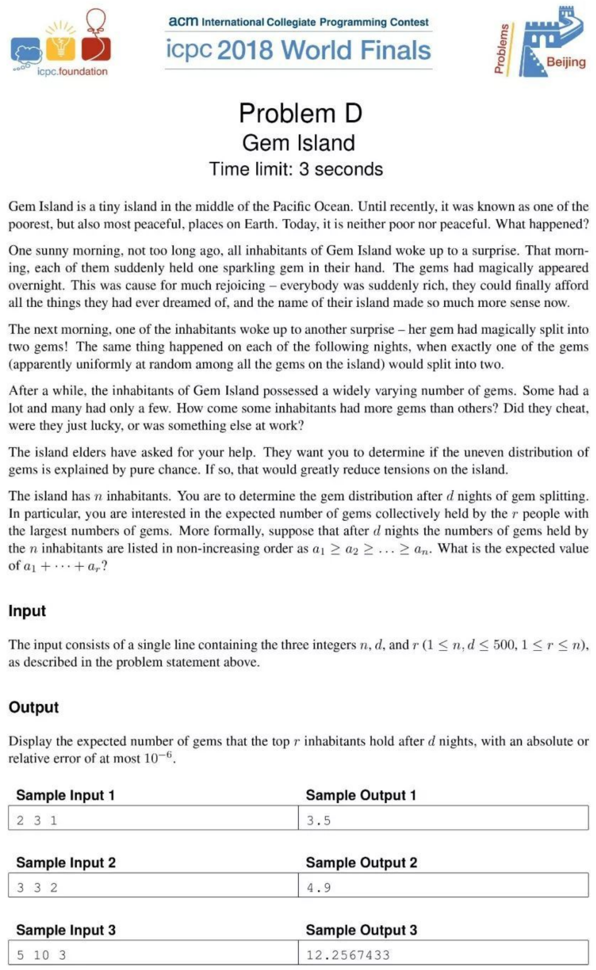
- Online Judges
  - Real contest problems
  - Immediate feedback
  - Codeforces, TopCoder. Leetcode, Hackerranks, Quera
- Periodic Practice Contests
  - University level communities
  - Local Coding Contests

Hamedan University of Technology

- Our Background
  - HUT-CPC
  - Achievements
- To Do
  - University level community (Programming G1, Programming G2, ACM-Course)
  - Weekly and Monthly Online Contests (Training)
  - Multiple Local Contests (Preparning)
  - HUT-Official Programming Contest (Qualification for Hamedan Programming Contest)
  - Hamedan-Official Programming Contest (Qualification for ACM-ICPC Sharif)

Programming Contest Problems Structure

- Title
- Limitations (Memory and RunTime)
- IO Files (Optional)
- Problem Description
- Input Structure and Limitations
- Output Structure
- Samples


acm International Collegiate Programming Contest

icpc 2018 World Finals

Problems Beijing

icpc.foundation

Problem D
Gem Island
Time limit: 3 seconds

Gem Island is a tiny island in the middle of the Pacific Ocean. Until recently, it was known as one of the poorest, but also most peaceful, places on Earth. Today, it is neither poor nor peaceful. What happened?

One sunny morning, not too long ago, all inhabitants of Gem Island woke up to a surprise. That morning, each of them suddenly held one sparkling gem in their hand. The gems had magically appeared overnight. This was cause for much rejoicing – everybody was suddenly rich, they could finally afford all the things they had ever dreamed of, and the name of their island made so much more sense now.

The next morning, one of the inhabitants woke up to another surprise – her gem had magically split into two gems! The same thing happened on each of the following nights, when exactly one of the gems (apparently uniformly at random among all the gems on the island) would split into two.

After a while, the inhabitants of Gem Island possessed a widely varying number of gems. Some had a lot and many had only a few. How come some inhabitants had more gems than others? Did they cheat, were they just lucky, or was something else at work?

The island elders have asked for your help. They want you to determine if the uneven distribution of gems is explained by pure chance. If so, that would greatly reduce tensions on the island.

The island has n inhabitants. You are to determine the gem distribution after d nights of gem splitting. In particular, you are interested in the expected number of gems collectively held by the r people with the largest numbers of gems. More formally, suppose that after d nights the numbers of gems held by the n inhabitants are listed in non-increasing order as $a_1 \geq a_2 \geq \ldots \geq a_n$. What is the expected value of $a_1 + \cdots + a_r$?

Input

The input consists of a single line containing the three integers n, d, and r ($1 \leq n, d \leq 500$, $1 \leq r \leq n$), as described in the problem statement above.

Output

Display the expected number of gems that the top r inhabitants hold after d nights, with an absolute or relative error of at most 10^{-6}.

Sample Input 1	Sample Output 1
`2 3 1`	`3.5`

Sample Input 2	Sample Output 2
`3 3 2`	`4.9`

Sample Input 3	Sample Output 3
`5 10 3`	`12.2567433`

Solution Flow

1. Read the problem statement
   - Check the input/output specification!
2. Make the problem abstract
3. Design an algorithm
   - Often the hardest step
4. Implement and debug
5. Submit
6. AC!
   - If not, go back to 4

Problem Solving Example

- A+B Problem
  - Inputs
    - Two space-separated integers a, b
  - Constraints
    - $0 \leq a, b \leq 10$
  - Output
    - $a + b$

Problem Solving Example

- A+B Problem
  - Inputs
    - Two space-separated integers a, b
  - Constraints
    - $0 \le a, b \le 10$
  - Output
    - a + b

```
#include<stdio.h>
int main()
{
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d\n", a + b);
    return 0;
}
```

Another Example

- Financial Management
  - Inputs
    - 12 floating point numbers on separate lines
  - Output
    - Average of the given numbers
  - Constrains?

Another Example

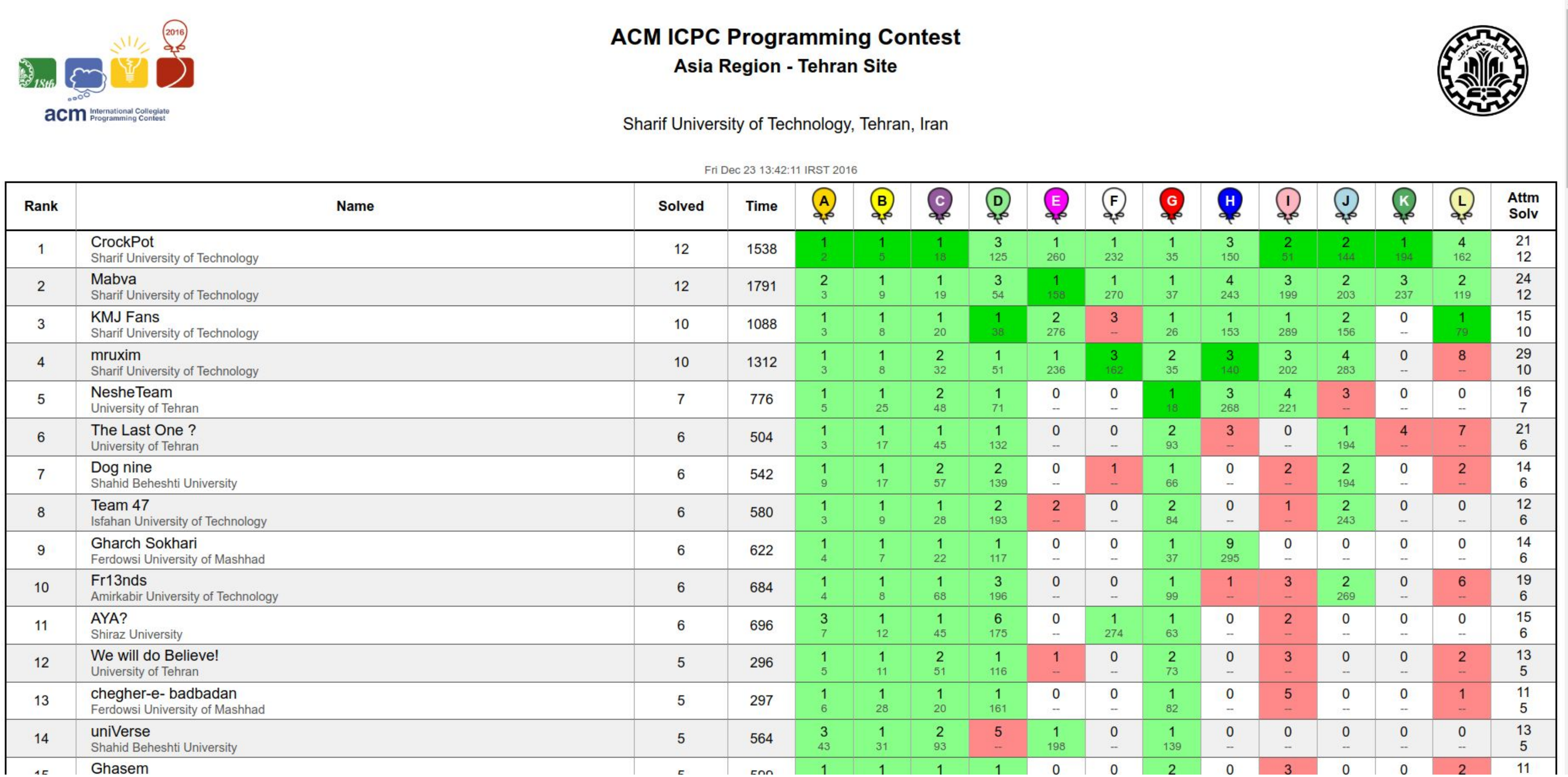
- Financial Management
  - Inputs
    - 12 floating point numbers on separate lines
  - Output
    - Average of the given numbers
  - Constrains?

```
#include<stdio.h>
int main() {
    double sum = 0, buf;
    for(int i = 0; i < 12; i++) {
        scanf("%lf", &buf);
        sum += buf;
    }
    printf("$%.2lf\n", sum / 12.0);
    return 0;
}
```

Another Example

- Financial Management
  - Inputs
    - 12 floating point numbers on separate lines
  - Output
    - Average of the given numbers
  - Constrains?
- Something to think about
  - What if the given numbers are HUGE?
  - Not all the input constraints are explicit
    - Hidden constraints are generally “reasonable”
  - Always think about the worst case scenario, edge cases, etc.

Scoreboard



ACM ICPC Programming Contest

Asia Region - Tehran Site

Sharif University of Technology, Tehran, Iran

Fri Dec 23 13:42:11 IRST 2016

Rank	Name	Solved	Time	A	B	C	D	E	F	G	H	I	J	K	L	Attm Solv
1	CrockPot Sharif University of Technology	12	1538	1 2	1 5	1 18	3 125	1 260	1 232	1 35	3 150	2 51	2 144	1 194	4 162	21 12
2	Mabva Sharif University of Technology	12	1791	2 3	1 9	1 19	3 54	1 158	1 270	1 37	4 243	3 199	2 203	3 237	2 119	24 12
3	KMJ Fans Sharif University of Technology	10	1088	1 3	1 8	1 20	1 38	2 276	3 --	1 26	1 153	1 289	2 156	0 --	1 79	15 10
4	mruxim Sharif University of Technology	10	1312	1 3	1 8	2 32	1 51	1 236	3 162	2 35	3 140	3 202	4 283	0 --	8 --	29 10
5	NesheTeam University of Tehran	7	776	1 5	1 25	2 48	1 71	0 --	0 --	1 18	3 268	4 221	3 --	0 --	0 --	16 7
6	The Last One ? University of Tehran	6	504	1 3	1 17	1 45	1 132	0 --	0 --	2 93	3 --	0 --	1 194	4 --	7 --	21 6
7	Dog nine Shahid Beheshti University	6	542	1 9	1 17	2 57	2 139	0 --	1 --	1 66	0 --	2 --	2 194	0 --	2 --	14 6
8	Team 47 Isfahan University of Technology	6	580	1 3	1 9	1 28	2 193	2 --	0 --	2 84	0 --	1 --	2 243	0 --	0 --	12 6
9	Gharch Sokhari Ferdowsi University of Mashhad	6	622	1 4	1 7	1 22	1 117	0 --	0 --	1 37	9 295	0 --	0 --	0 --	0 --	14 6
10	Fr13nds Amirkabir University of Technology	6	684	1 4	1 8	1 68	3 196	0 --	0 --	1 99	1 --	3 --	2 269	0 --	6 --	19 6
11	AYA? Shiraz University	6	696	3 7	1 12	1 45	6 175	0 --	1 274	1 63	0 --	2 --	0 --	0 --	0 --	15 6
12	We will do Believe! University of Tehran	5	296	1 5	1 11	2 51	1 116	1 --	0 --	2 73	0 --	3 --	0 --	0 --	2 --	13 5
13	chegher-e- badbadan Ferdowsi University of Mashhad	5	297	1 6	1 28	1 20	1 161	0 --	0 --	1 82	0 --	5 --	0 --	0 --	1 --	11 5
14	uniVerse Shahid Beheshti University	5	564	3 43	1 31	2 93	5 --	1 198	0 --	1 139	0 --	0 --	0 --	0 --	0 --	13 5
15	Ghasem	5	[illegible]	1	1	1	1	0	0	2	0	3	0	0	2	11

Balloon and Problem Color

- Why?
- When?
- Where?

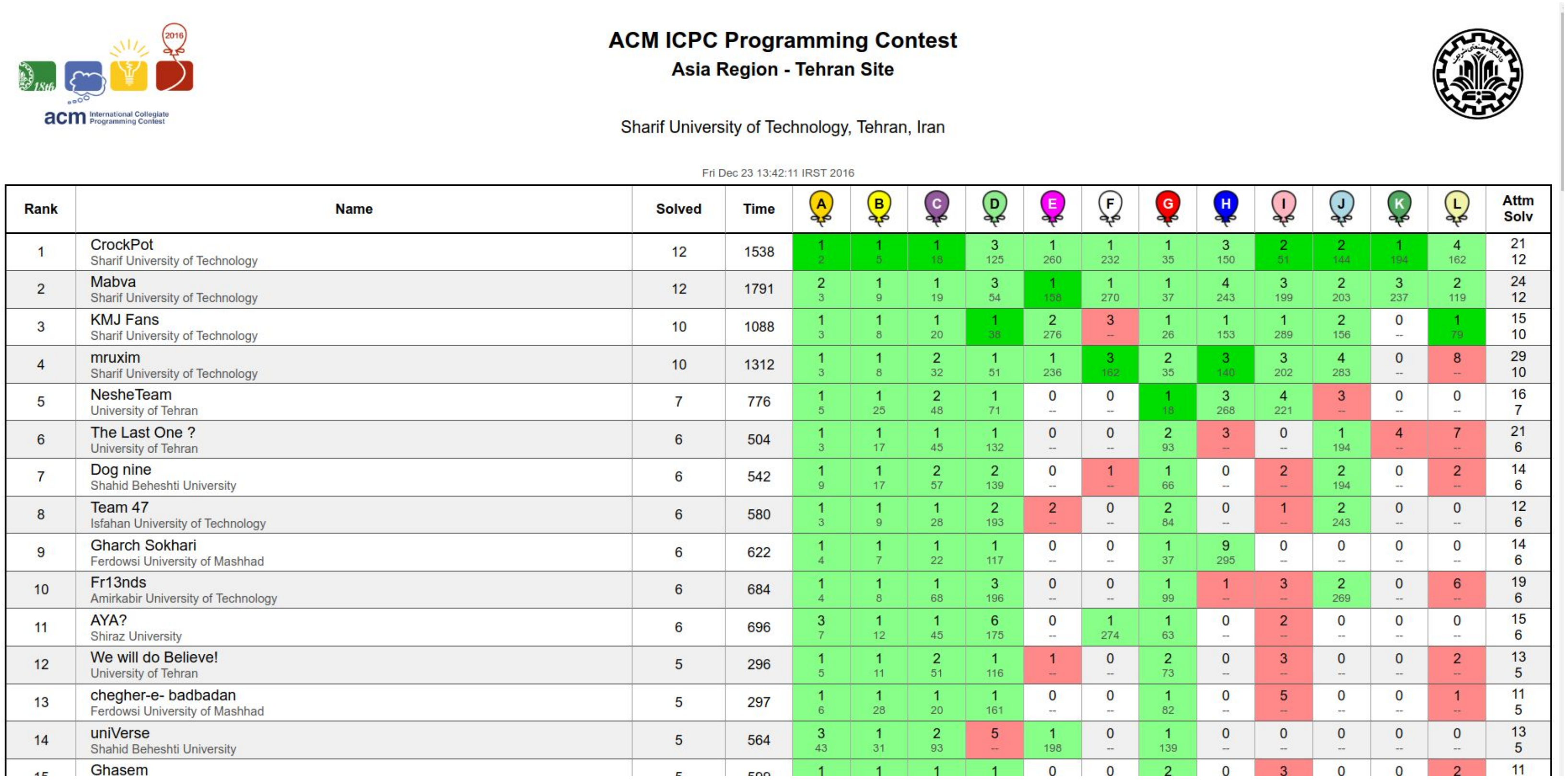
Explanation of the scoring and scoreboard in ACM-ICPC contests

- Rank
  - Highest number of solved problems.
  - Least score within the same number of solved problems.
  - Fastest submission time for the latest correct submission, as a tie-breaker.
- Name
  - University logo, country flag and team name.
- Solved
  - Number of solved problems.
- Score
  - Total score: The sum of the fastest solution time plus a penalty of 20 minutes per incorrect attempt for all solved problems.
  - If a problem is not solved, the attempts do not give any penalty to the score.

Explanation of the scoring and scoreboard in ACM-ICPC contests

- A-Z
  - Problems:
    - The problem is neither attempted nor solved.
    - 3 · There have been 3 attempts, but the problem remains unsolved.
    - 4/122 The problem was solved on the fourth attempt after 122 minutes.
- Total att/solv
  - Total submission attempts/number of solved problems
- Change
  - Recent change in ranking during the contest.

Scoreboard



ACM ICPC Programming Contest

Asia Region - Tehran Site

Sharif University of Technology, Tehran, Iran

Fri Dec 23 13:42:11 IRST 2016

Rank	Name	Solved	Time	A	B	C	D	E	F	G	H	I	J	K	L	Attm Solv
1	CrockPot Sharif University of Technology	12	1538	1 2	1 5	1 18	3 125	1 260	1 232	1 35	3 150	2 51	2 144	1 194	4 162	21 12
2	Mabva Sharif University of Technology	12	1791	2 3	1 9	1 19	3 54	1 158	1 270	1 37	4 243	3 199	2 203	3 237	2 119	24 12
3	KMJ Fans Sharif University of Technology	10	1088	1 3	1 8	1 20	1 38	2 276	3 --	1 26	1 153	1 289	2 156	0 --	1 79	15 10
4	mruxim Sharif University of Technology	10	1312	1 3	1 8	2 32	1 51	1 236	3 162	2 35	3 140	3 202	4 283	0 --	8 --	29 10
5	NesheTeam University of Tehran	7	776	1 5	1 25	2 48	1 71	0 --	0 --	1 18	3 268	4 221	3 --	0 --	0 --	16 7
6	The Last One ? University of Tehran	6	504	1 3	1 17	1 45	1 132	0 --	0 --	2 93	3 --	0 --	1 194	4 --	7 --	21 6
7	Dog nine Shahid Beheshti University	6	542	1 9	1 17	2 57	2 139	0 --	1 --	1 66	0 --	2 --	2 194	0 --	2 --	14 6
8	Team 47 Isfahan University of Technology	6	580	1 3	1 9	1 28	2 193	2 --	0 --	2 84	0 --	1 --	2 243	0 --	0 --	12 6
9	Gharch Sokhari Ferdowsi University of Mashhad	6	622	1 4	1 7	1 22	1 117	0 --	0 --	1 37	9 295	0 --	0 --	0 --	0 --	14 6
10	Fr13nds Amirkabir University of Technology	6	684	1 4	1 8	1 68	3 196	0 --	0 --	1 99	1 --	3 --	2 269	0 --	6 --	19 6
11	AYA? Shiraz University	6	696	3 7	1 12	1 45	6 175	0 --	1 274	1 63	0 --	2 --	0 --	0 --	0 --	15 6
12	We will do Believe! University of Tehran	5	296	1 5	1 11	2 51	1 116	1 --	0 --	2 73	0 --	3 --	0 --	0 --	2 --	13 5
13	chegher-e- badbadan Ferdowsi University of Mashhad	5	297	1 6	1 28	1 20	1 161	0 --	0 --	1 82	0 --	5 --	0 --	0 --	1 --	11 5
14	uniVerse Shahid Beheshti University	5	564	3 43	1 31	2 93	5 --	1 198	0 --	1 139	0 --	0 --	0 --	0 --	0 --	13 5
15	Ghasem	5	599	1	1	1	1	0	0	2	0	3	0	0	2	11

Notebooks

- Models
  - Team:
    - No Cool Disk or Internet Access
  - Host
    - Notebook if needed
    - Available Some Blank Papers and Pen
- Exceptions
  - Based on Programming Contest Policies

Links

- Official Host Website: icpc.hut.ac.ir
- Training Online Course: https://quera.org/course/15629/
    - Online Programming Contests:
- Official Telegram Channel: @hut-icpc
- Some useful courses
    - To do:
    - Codeforces, Quera, …
- Archives
    - Coming soon

Discussion and FAQ!

- Local Contests
- Editorial
- Plans
- ...

Problem TB1 (Beginner Level)

Financial Management

Time Limit: 1000MS	Memory Limit: 10000K
Total Submissions: 239034	Accepted: 88806

Description

Larry graduated this year and finally has a job. He's making a lot of money, but somehow never seems to have enough. Larry has decided that he needs to grab hold of his financial portfolio and solve his financing problems. The first step is to figure out what's been going on with his money. Larry has his bank account statements and wants to see how much money he has. Help Larry by writing a program to take his closing balance from each of the past twelve months and calculate his average account balance.

Input

The input will be twelve lines. Each line will contain the closing balance of his bank account for a particular month. Each number will be positive and displayed to the penny. No dollar sign will be included.

Output

The output will be a single number, the average (mean) of the closing balances for the twelve months. It will be rounded to the nearest penny, preceded immediately by a dollar sign, and followed by the end-of-line. There will be no other spaces or characters in the output.

Problem TB1 (Beginner Level)

Financial Management

Time Limit: 1000MS	Memory Limit: 10000K
Total Submissions: 239034	Accepted: 88806

Sample Input 1

```
100.00
489.12
12454.12
1234.10
823.05
109.20
5.27
1542.25
839.18
83.99
1295.01
1.75
```

Sample Output 1

```
$1581.42
```

Problem TB1 (Beginner Level)

Financial Management

Time Limit: 1000MS	Memory Limit: 10000K
Total Submissions: 239034	Accepted: 88806

Sample Input 1

```
100.00
489.12
12454.12
1234.10
823.05
109.20
5.27
1542.25
839.18
83.99
1295.01
1.75
```

```python
sum = 0.0
for i in range(12):
    sum += float(input())

print(f'${round(sum/12.0, 2)}')
```

Sample Output 1

```
$1581.42
```

Problem TB2 (Beginner Level)

Vertical Histogram

Time Limit: 1000MS	Memory Limit: 65536K
Total Submissions: 22198	Accepted: 10429

Description

Write a program to read four lines of upper case (i.e., all CAPITAL LETTERS) text input (no more than 72 characters per line) from the input file and print a vertical histogram that shows how many times each letter (but not blanks, digits, or punctuation) appears in the all-upper-case input. Format your output exactly as shown.

Input

* Lines 1..4: Four lines of upper case text, no more than 72 characters per line.

Output

* Lines 1..??: Several lines with asterisks and spaces followed by one line with the upper-case alphabet separated by spaces. Do not print unneeded blanks at the end of any line. Do not print any leading blank lines.

Problem TB2 (Beginner Level)

Vertical Histogram

Time Limit: 1000MS	Memory Limit: 65536K
Total Submissions: 22198	Accepted: 10429

Sample Input 1

```
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOG.
THIS IS AN EXAMPLE TO TEST FOR YOUR
HISTOGRAM PROGRAM.
HELLO!
```

Sample Output 1

```
                            *
                            *
        *                   *
        *                   *     *   *
        *                   *     *   *
*       *     *             *     *   *
*       *     * *     * *   *     * * *
*       *   * * *     * *   * *   * * * *
*     * * * * * *     * * * * *   * * * *     * *
* * * * * * * * * * * * * * * * * * * * * * * * * *
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

Problem TB2 (Beginner Level) | Vertical Histogram | Solution

```cpp
#include<iostream>
#include<string>
#include<algorithm>
#include<vector>
#include<cmath>

using namespace std;

int main(){

  int histogram[26] = {0,};
  int N=4;
  string tmp;
  while(N--){
    getline(cin,tmp);
    for(int i=0;i<tmp.size();i++){
      if(tmp[i] < 'A' && tmp[i] > 'Z')
        continue;
      histogram[tmp[i]-'A']++;
    }
  }
  int max=0;

  for(int i=0;i<26;i++){
    if(max < histogram[i])
      max=histogram[i];
  }

  for(int i=max;i>0;i--){
    for(int j=0;j<26;j++){
      if(j < 25){
        if(histogram[j] >= i)
          cout<<"* ";
        else
          cout<<"  ";
      }
      else{
        if(histogram[j] >= i)
          cout<<"*";
        else
          cout<<" ";
      }
    }
      cout<<endl;
  }
  for(int j=0;j<26;j++){

    cout<<(char)('A'+j);
    if(j<25)
      cout<<" ";
  }
    cout<<endl;
    return 0;
}
```

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Each team at the World Finals will be provided with one workstation, shared between team members.
- Note that this is the traditional ICPC setup (not the unique setup of some previous World Finals where each team member was provided with a separate workstation).
- Configuration (coming slides)

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Hardware
  - System Unit
    - Dell OptiPlex 7090 – Tower version , with 16GB RAM, 128GB SSD, and Intel(R) Core(TM) i7-10700 processor at 2.90GHz.
  - External screen:
    - Dell Model E2421HN 23.8 inch, 1920×1080, W-LED monitor
  - External keyboard:
    - Dell Model KB216 wired keyboard (a keyboard image is posted here)
  - External Mouse
    - Dell Optical Wired Mouse Model MS116 (a mouse image is posted here)
  - Webcam:
    - To Be Determined (The webcam will be used to capture team member images during the entire contest, and may not be redirected or blocked).

No hardware substitutions will be allowed (that is, teams may not bring their own equipment onto the contest floor). This includes that teams may not substitute keyboards or other peripherals; all teams will use identical equipment, as described above, during the contest.

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Software
  - OS:
    - Ubuntu 22.04.1 LTS Linux (64-bit).
    - A list of packages included in the OS image is available
  - Desktop:
    - GNOME
  - Editors
    - vi/vim
    - Gvim
    - Emacs
    - Gedit
    - Geany
    - kate

ICPC World Finals Programming Environment (Update: 46th & 47th)

- IDEs:
    - Eclipse (version 2022-12), configured with:
        - Java Development Tooling (JDT) version 3.18.1400.v20221123-1800 using Java as listed above.
        - C++ Development Tooling (CDT) version 11.0.0 using C++ as listed above.
        - Python Development Tooling (PyDev) version 10.0.2 using Python3 as listed above.
    - IntelliJ (IDEA Community Edition, version 2022.3), configured with:
        - Java as listed above
        - Kotlin as listed above
    - CLion (version 2022.3), configured with:
        - C/C++ as listed above
    - Pycharm Community Edition Python IDE (version 2022.3), configured with:
        - Python 3 as listed above
    - Code::Blocks (version 20.03-3.1), configured with:
        - C/C++ as listed above
    - VS Code(version 1.74.2 configured with Microsoft C/C++ extension V1.13.8)
- NOTE: the Judges will compile and execute C/C++ programs using C/C++ as listed under Languages, above, NOT using Microsoft C/C++.

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Languages:
    - Java (Openjdk version 17.0.5 2022-10-18)
    - C (gcc 11.3.0 (Ubuntu 11.3.0-1ubuntu1~22.04))
    - C++ (g++ 11.3.0 (Ubuntu 11.3.0-1ubuntu1~22.04))
    - Python 3 (PyPy 7.3.10 with GCC 9.4.0 providing python 3.9.15) A list of the installed Python modules is available here.
    - Kotlin (Version 1.7.21)
  - Note that Python 2 is no longer supported at the World Finals.

The programming languages of the regional contest will include C and C++. Additional programming languages may be used. The programming languages of the ICPC World Finals are Java, Kotlin, Python and C/C++. although it is not guaranteed every problem is solvable in any certain language, the ICPC website states that "the judges will have solved all problems in Java and C++" for both regional and world finals competitions).

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Compilation of Submissions
- During the contest, teams will submit proposed solutions to the contest problems to the Judges using the DOMjudge contest control system (CCS). Source files submitted to the Judges will be compiled using the following command line arguments for the respective language:
    - C:
        - gcc -x c -g -O2 -std=gnu11 -static ${files} -lm
    - C++:
        - g++ -x c++ -g -O2 -std=gnu++20 -static ${files}
    - Java:
        - javac -encoding UTF-8 -sourcepath . -d . ${files}
    - Python 3
        - pypy3 -m py_compile ${files}
    - Kotlin
        - kotlinc -d . ${files}

ICPC World Finals Programming Environment (Update: 46th & 47th)

- The “${files}” in the above commands represents the list of source files from the submission which will actually be compiled. Files with the following suffixes (and only files with these suffixes) will be submitted to the compiler:
- File extensions:
    - For C submissions: files ending with .c
    - For C++ submissions: files ending with .cc, .cpp, .cxx, or .c++
    - For Java submissions: files ending with .java
    - For Python submissions: files ending with .py
    - For Kotlin submissions: files ending with .kt

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Execution of Submissions
- For each language, if the above compilation step is successful then the submission will be executed as follows:
    - For C/C++: the executable file generated by the compiler will be executed to generate the output of the submission.
    - For Python 3: the main source file will be executed by the PyPy3 Python3 interpreter to generate the output of the submission.
    - For Java: the compiled main class will be executed using the following command:
        - java -Dfile.encoding=UTF-8 -XX:+UseSerialGC -Xss64m -Xms1920m -Xmx1920m
    - For Kotlin: the compiled main class will be executed using the following command:
        - kotlin -Dfile.encoding=UTF-8 -J-XX:+UseSerialGC -J-Xss64m -J-Xms1920m -J-Xmx1920m
- Compilation and execution as described above will take place in a “sandbox” on a dedicated judging machine. The judging machine will be as identical as possible to, and at least as powerful as, the machines used by teams. The sandbox will allocate 2GB of memory; the entire program, including its runtime environment, must execute within this memory limit. For interpreted languages (Java, Python, and Kotlin) the runtime environment includes the interpreter (that is, the JVM for Java/Kotlin and the Python interpreter for Python).
- The sandbox memory allocation size will be the same for all languages and all contest problems. For Java and Kotlin, the above commands show the stack size and heap size settings which will be used when the program is run in the sandbox.

ICPC World Finals Programming Environment (Update: 46th & 47th)

- Reference Materials
  - The following packages will be available on team machines at the World Finals, and will be installed automatically as part of the steps listed under Building Your Own World Finals Machine, above.
    - JDK JavaDocs
    - C++ STL docs
    - DOMjudge Team Guide
    - Additional Information:
    - Judging Notes
    - Judging Notes Addendum
    - Technical Notes

ICPC Regional Finals Championships

- The regional contests are partitioned into eight championships; the ICPC Africa & Arab Championship, the ICPC Asia East Championship, the ICPC Asia Pacific Championship, the ICPC Asia West Championship, the ICPC European Championship, the ICPC Latin America Championship, the ICPC Northern Eurasia Championship, and the ICPC North America Championship. Additional Championships may be defined by the ICPC Executive Committee.
- The highest level Regional Contest may be a Championship Contest. Team composition and requirements rules for a Championship contest are to be the same as for the World Finals. A Championship contest should be held no later than three months prior to the World Finals.
- Note: The 2023/24 Regional Schedule may extend into the first quarter of 2024.

Advancing to the ICPC World Finals

- The highest level Regional Contests advance teams to the ICPC World Finals. Additional teams who competed in the highest level of regional contests may be invited to the ICPC World Finals as wild card teams.
- Teams qualify to advance to the ICPC World Finals through Regional Contests and by satisfying all rules posted in The Rules of the ICPC World Finals. Specifically:
- To qualify for the ICPC World Finals, the coach and all team members must be fully registered in the ICPC Registration System BEFORE competing in ANY regional qualifying event. Incomplete registration or circumvention that leads to incomplete or false data is grounds for immediate disqualification.
- Only one team from a given institution may advance to the ICPC World Finals. No team member on the qualifying team may have competed as a contestant in two previous ICPC World Finals.
- The coach of a qualifying team is the point-of-contact before and during ICPC World Finals activities. The coach must complete certification at the Team Certification Web Site within five (5) days of notification. Qualifying teams will be invited by email within one day of completing certification.
- Qualifying teams requiring visas must initiate the process of applying for visas within ten days of being issued an invitation. Teams failing to comply with any of these requirements will be ruled ineligible to compete in the ICPC World Finals. Upon completion of these requirements, a qualifying team will be advanced to the ICPC World Finals.
- A team advancing to the ICPC World Finals will be comprised of the same three members as when it qualified. If a team member is unwilling, unable or unfit to compete in the ICPC World Finals, the coach must notify the ICPC Manager promptly. A team member who is unwilling or unfit to compete in the ICPC World Finals will be disqualified from further ICPC competitions. The team member may appeal disqualification to the Appeals Committee. At on-site registration, participants must provide a picture ID (passport, drivers license, etc). Contestants must show proof of enrollment at the university during the term of the regional contest at which they qualified. A letter on university stationery with the signature of a university official accompanied by an English translation is sufficient.

Regional Contest Computing Environment

- The programming languages of the regional contest will include C and C++ and JAVA. Additional programming languages may be used. The programming languages of the ICPC World Finals are Java, Kotlin, Python and C/C++.
- Each team will use a single workstation. The regional contest director is responsible for determining that teams have reasonably equivalent computing resources.
- Each Regional Contest Director determines whether contestants may bring materials for use during the contest. Please see the specific regional rules at the ICPC Regional Contest Web Site.
- At the ICPC World Finals, no printed materials or electronic devices may be brought into the contest area. On-line reference materials will be made available as described in the Programming Environment Web Site. Each team will be permitted to provide a PDF of up to 25 pages of notes within the limits described during Team Certification. Three copies will be printed and placed at the team's workstation for use during the ICPC World Finals.

Who all can participate?

- The eligibility requirement of the ACM ICPC contest is:
- Any student, who is good in computer programming language and is willing to participate.
- The student should be enrolled with some specific institute for a degree programme.
- The student can only represent one institute in a specific calendar period.
- The student should not have been a part of the two contest finals.
- The student should not have been a part of 5 contest regionals.
- The participant cannot be a part of more than two teams in the same calendar year.
- The participant cannot take part in more than two regionals in the same calendar year.

How to prepare for ICPC?

- Select a computer language that suits you the best
- For acing the ACM-ICPC competition, a basic knowledge of any computer programming language will not work. You will have to have advanced knowledge of at least one computer language to solve the problems quickly.
- If you are already well versed in a specific computer language like C++, C, or Python, then practice it well. If you are a beginner, then pick one language and learn it thoroughly. It is needless to say that picking the correct language for you is quintessential in this case.

Learn concepts thoroughly

- When you start practising for ICPC, do not leave any concept. Every concept is essential as there is no prescribed question pattern. When you learn advanced algorithms, you will understand that skipping any topic is not an option since each topic is correlated.
- One of the best ways to identify your progress and mistakes is to test yourself. Try to take some online coding tests to analyse your preparation. With the help of online coding tests, you can quickly identify your strengths and weaknesses and analyse your score with your fellow competitors.
- It would be best to go through questions that were a part of the previous years' ICPC assessment to understand the complexity and level of problems. The programming aptitude questions are a little tough to understand but with great practice, they can be solved. This way, you will have a clear idea about your preparation. In addition, you can locate the past problems from ACM-ICPC archives.

Practice till you win it

- Practice is the key to winning. For every competitive exam, practice can only make you perfect. Solving more complicated problems, concentrating on areas that you find difficult, and understanding every concept are a must.
- Practice by taking various online tests and quizzes. Some online tests for programming languages declare results with your rank, which is a great way to analyse your performance with your peers.

Building a great team

- Since ACM-ICPC is a team-based contest, building a good team is also very important. Select teammates who are flexible and have the same level of enthusiasm and commitment as you do. This will help in keeping everyone motivated and focused on the goal.
- Select a diverse team so that the team can solve the different problems in the competition. The team members should be willing to make efforts to achieve the final goal.
- The most important thing is to practice with your team. Every member of the team should know each member’s strengths and weaknesses and complement each other.

Frequently Asked Questions

- Can a graduate apply for this contest?
  - Yes, a person who has gained a degree in computer programming or language can apply for this contest, but they should be enrolled under an institution.
- Are ACM-ICPC conducted for individuals too?
  - ACM-ICPC contest is a team-based contest with three members in each team. So, the ACM-ICPC contest is not an individual contest.
- When will the regional round of the contest happen?
  - The regional round of the ACM-ICPC contest will likely take place in the second week of July.

Key Takeaways

- The ACM-ICPC is one of the biggest and oldest international computer programming contests that is carried on a multi-layered level. It is a team contest, consisting of 3 members in each team. Students who have a keen interest in computer programming language and can work great in a team should participate.
- Students currently enrolled in a degree programme in a specific institute are eligible to participate in this contest. If the regional rounds are cleared, the winning team will represent its institute and country internationally.
- Being a part of this contest will surely open so many opportunities for you. To win such an elite competition, you will have to be well prepared. With practice, your skill set increases, and it will help you find good online and off-campus placements.

How to Practice

- Regional and International Official Problems Archive
- Quora Environment
    - Courses (My Course, HUT Programming Contest Preparing Course)
    - Archived Problems (for previous programming contests)
    - Online Contests
    - Example of Submission
- Codeforces
    - Example of Submission
- Leetcode
- TopCoders
- ...

How to Practice

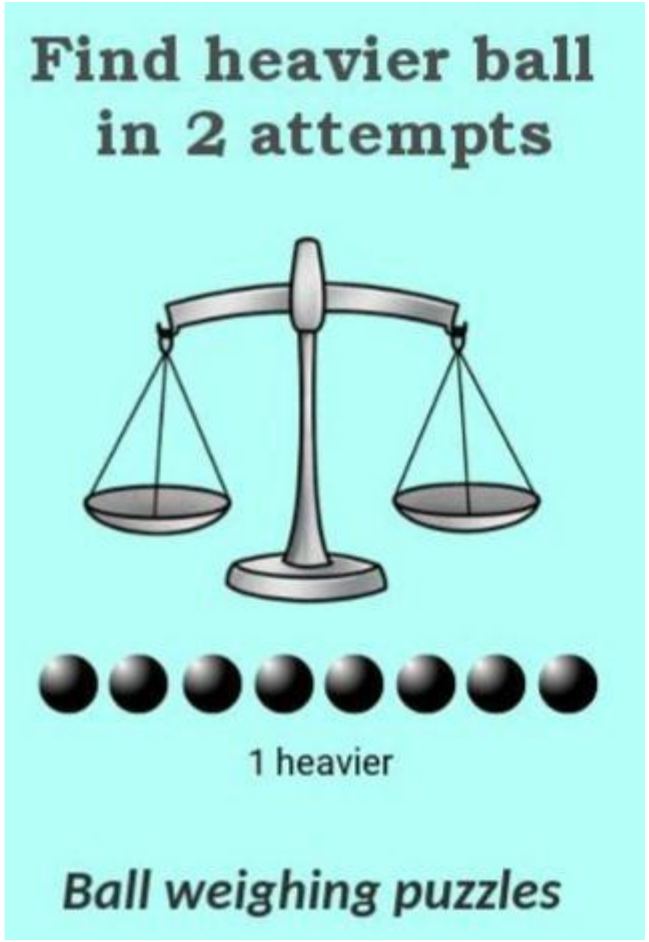
- Regional and International Official Problems Archive
- Quora Environment
  - Courses (My Course, HUT Programming Contest Preparing Course)
  - Archived Problems (for previous programming contests)
  - Online Contests
  - Example of Submission
- Codeforces
  - Example of Submission
- Leetcode
- TopCoders
- ...

Topics

- Introduction
- <u>Algorithm Complexity</u>
- Mathematics
- Data structures
- Dynamic programming (DP)
- Combinatorial games
- Graph algorithms
- Shortest distance problems
- Network flow
- Geometric algorithms
- String algorithms

Solution Vision

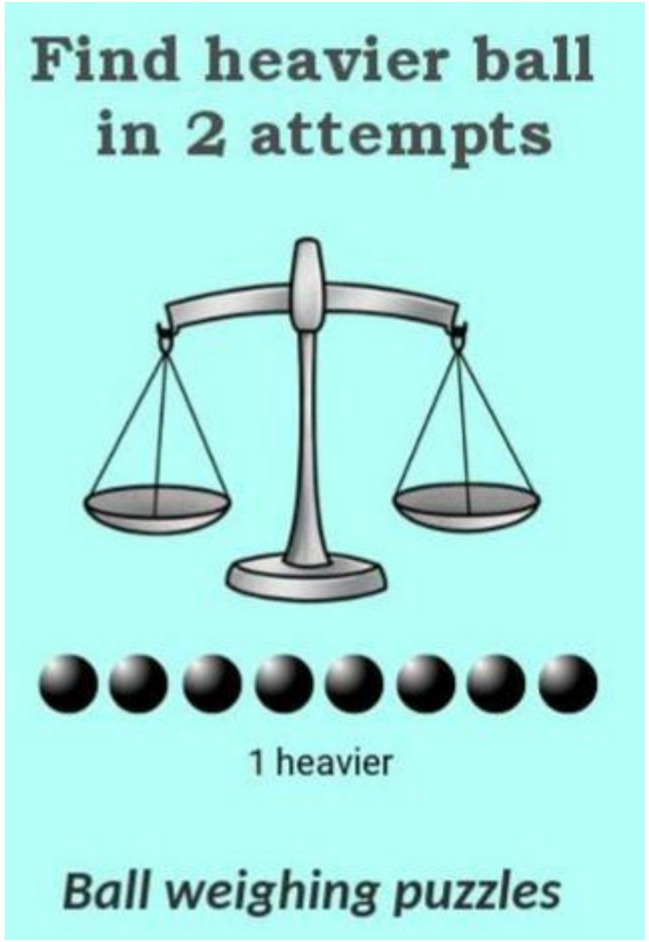
Ball Weighing Puzzle



Solution Vision

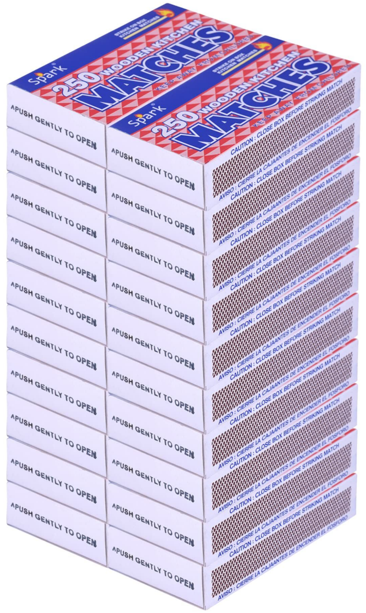
Ball Weighing Puzzle

- Incorrect Starting Point
- Different Vision is needed
- Human Brain Mistakes



Solution Vision

We have 20 boxes and each contains 250 matches. 9 boxes have matches of 1 gram and in 1 box we have matches of 1.1 grams. So if we have a digital weighing machine and we get to weigh only once to find the different box, how is it possible?



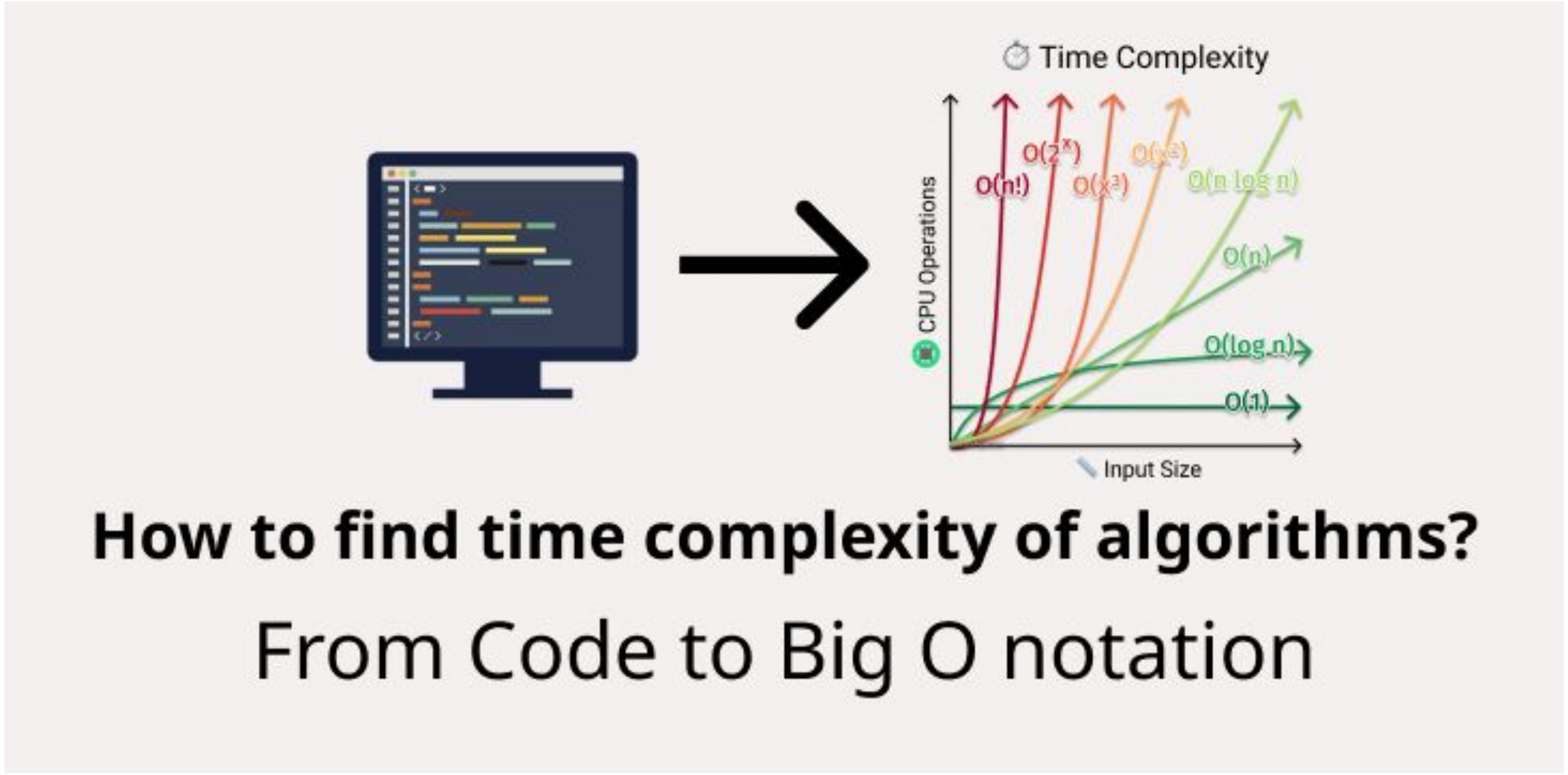
Solution Vision

- Think Different
  - Skip problem
  - On other starting point
- Don't use same vision for other problem (in most cases)

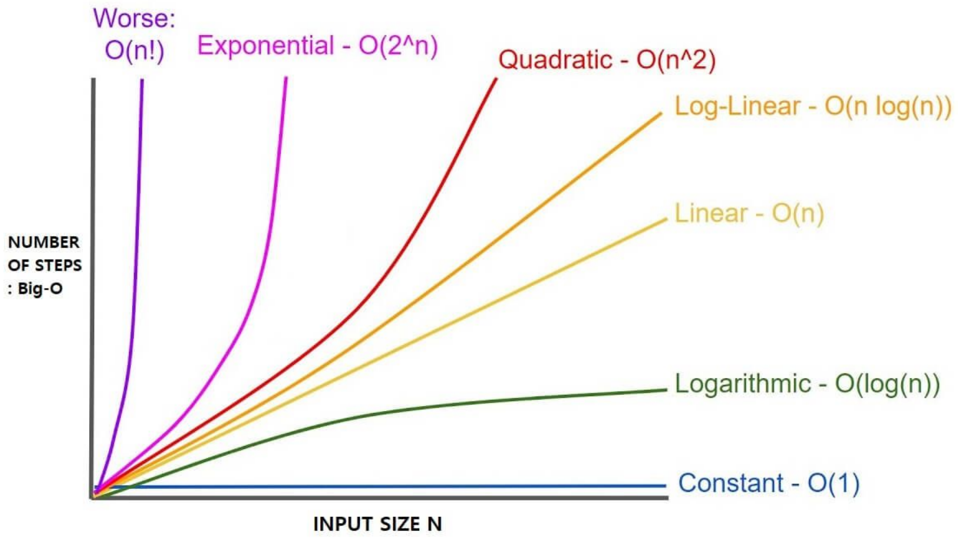
Algorithm Complexity

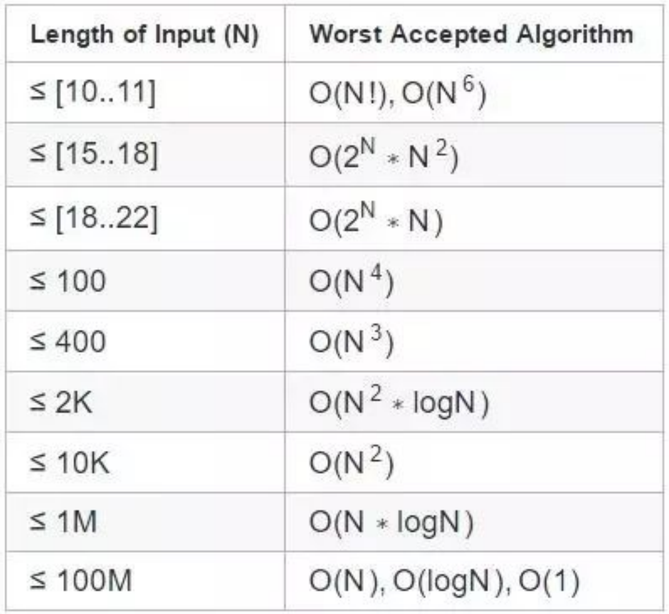
- Resources Complexity
  - Memory Usage
- Execution Complexity
  - Advantages
  - Disadvantages
- Algorithms Complexity
  - Advantages
  - Disadvantages

Algorithm Complexity



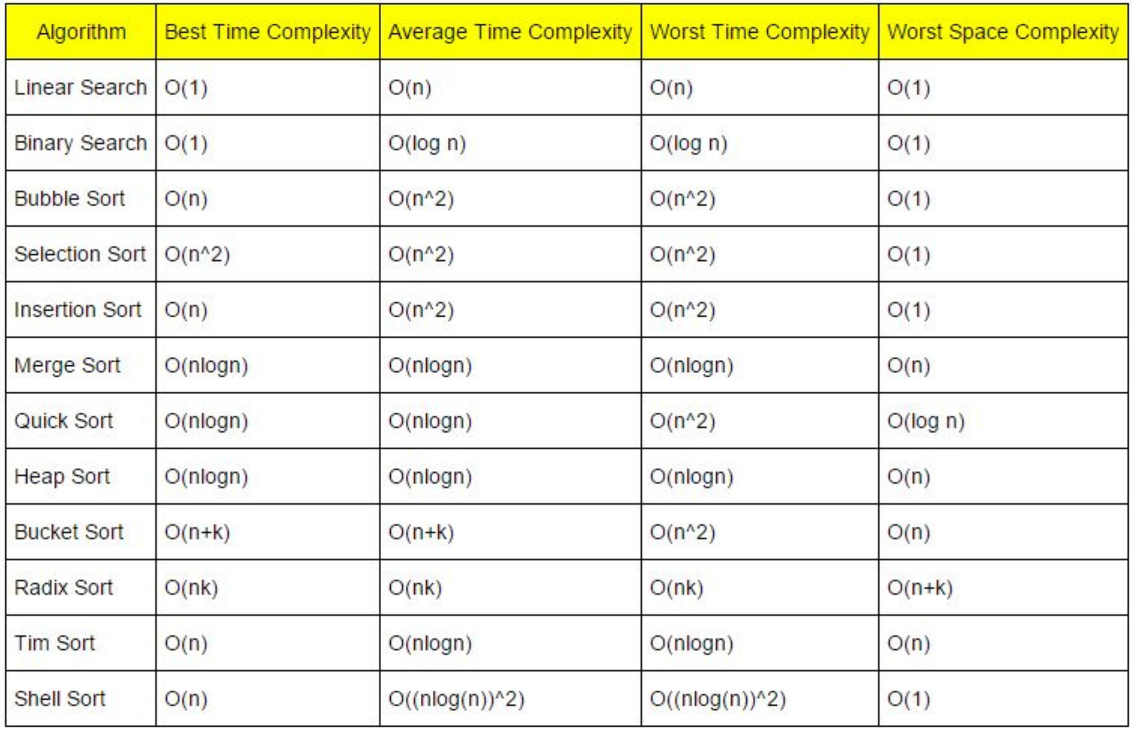
Algorithm Complexity





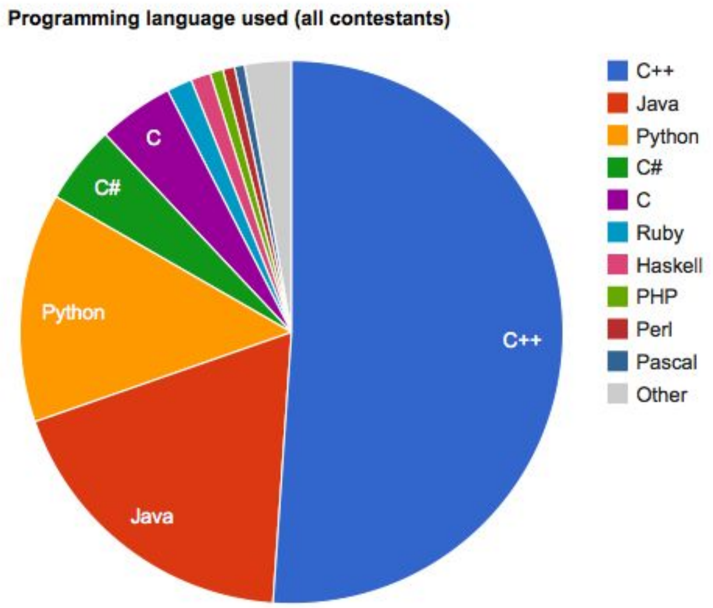
Length of Input (N)	Worst Accepted Algorithm
≤ [10..11]	$O(N!), O(N^6)$
≤ [15..18]	$O(2^N * N^2)$
≤ [18..22]	$O(2^N * N)$
≤ 100	$O(N^4)$
≤ 400	$O(N^3)$
≤ 2K	$O(N^2 * \log N)$
≤ 10K	$O(N^2)$
≤ 1M	$O(N * \log N)$
≤ 100M	$O(N), O(\log N), O(1)$

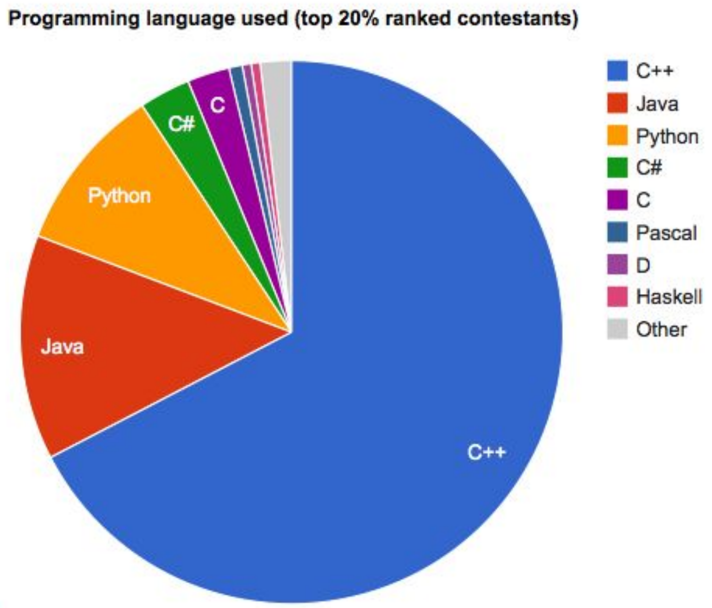
Searching/Sorting Algorithm Complexity



Algorithm	Best Time Complexity	Average Time Complexity	Worst Time Complexity	Worst Space Complexity
Linear Search	O(1)	O(n)	O(n)	O(1)
Binary Search	O(1)	O(log n)	O(log n)	O(1)
Bubble Sort	O(n)	O(n^2)	O(n^2)	O(1)
Selection Sort	O(n^2)	O(n^2)	O(n^2)	O(1)
Insertion Sort	O(n)	O(n^2)	O(n^2)	O(1)
Merge Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Quick Sort	O(nlogn)	O(nlogn)	O(n^2)	O(log n)
Heap Sort	O(nlogn)	O(nlogn)	O(nlogn)	O(n)
Bucket Sort	O(n+k)	O(n+k)	O(n^2)	O(n)
Radix Sort	O(nk)	O(nk)	O(nk)	O(n+k)
Tim Sort	O(n)	O(nlogn)	O(nlogn)	O(n)
Shell Sort	O(n)	O((nlog(n))^2)	O((nlog(n))^2)	O(1)

Programming Language Used by Contestants (ICPC and other Contests)





Thanks for your attention