PROGRAMMING CLUB

Week 1 – implementation

Oct 18th, 2016

By Yizuo Chen, Yiyou Chen

- 1. language (C++, java, python)
- 2. implementation
- 3. problems of implementation

LANGUAGE

- Language difference does not matter your ability to solve a problem
- Dividing groups by languages(3 people per team)
- Don’t know any language? Learn one. Code academy…
- If you have question about grammar, you could also ask on Onenote and we will help you.

IDEONE.COM

- http://ideone.com

IMPLEMENTATION(AD HOC PROBLEMS)

- Understand the problem.
- Do what problems ask.
- Sometimes little thinking.
- It can be easy, but it can also be hard.

WHAT?

- `Ad hoc' problems are those whose algorithms do not fall into standard categories with well-studied solutions. Each ad hoc problem is different; no specific or general techniques exist to solve them.
- Ad hoc problems usually require careful reading and usually yield to an attack that revolves around carefully sequencing the instructions given in the problem.

PROBLEM 1– ICPC #1FROM LAST YEAR'S CONTEST

Problem discription:

"I once met an interesting guy who had a phone with two batteries." - Dudu, 2014

Dudu once met an interesting chap who had a phone with two batteries. With that in mind he decided to create a phone that would have a pair of batteries for his upcoming trip to Thailand.

His company, Interesting Chap Phone Charging (ICPC), created an advanced dual battery technology. While one battery is being used to supply the phone the other is being recharged via a solar receptor.

Unfortunately the rate that the phone uses energy is higher than the rate a battery can be recharged. ICPC is interested to know how long a system will last until being completely out of charge.

PROBLEM 1 – ICPC

Input Format

The input will consist of 4 integers A, B, C and D. They are respectively the rate a phone uses energy in Amperes (Coulombs per second), the rate in which a battery can be recharged in Amperes, the initial charge of the first battery in Coulombs and the initial charge of the second battery in Coulombs.

ICPC's batteries have infinite capacity and the phone changes from using one battery to using the other instantaneously.

Constraints

$0 \leq A,B,C,D < 231$
$A > B$

Output Format

Output one integer with the number of seconds a phone will last given an ICPC dual battery as described in the statement. Round down your answer in case it's nonintegral.

Sample Input

```
5 1 10 10
```

Sample Output

```
5
```

PROBLEM 1 -- ICPC

- problem:

Two batteries. Always one works and the other recharges. We know the initial charges of both batteries(C and D) and the rate of using(A) and recharging(B). How long can they last?

- idea:

Always one works and one recharges. Every unit time uses up (A – B) energy.

Initially total (C + D) energy.

t = total / v

PROBLEM 2 --UNIT CONVERSION(LAST YEARS CONTEST)

- **"How do you say 'two kilos' in English?** -- Dudu, 2015
- Dudu always had trouble with measurements and unit conversions. Can you help him?
- **Input Format**
- The input will begin with a line containing 2 numbers x and y meaning that "x of unit A" is equal to "y of unit B". For instance, if A is "kilos" and B is "pounds" one possibility is x = 3.25 and y = 7.165024.
- The next line will contain a single integer N, containing the number of conversions to be performed.
- Each of the next N lines will be of the form "z q" where z is a number and q is either 'A' or 'B'.

- **Constraints**
- 1 ≤ N ≤ 100000
  x,y are strictly positive numbers
  z is a nonnegative number
- **Output Format**
- Output N lines with a number each. See the sample input/output for further details.
- **Sample Input**
- 3.25 7.165024
  5
  3.25 A
  1 A
  0 B
  2.1 B
  0 A
- Sample Output
- 7.165024
- 2.20462277
- 0.000000000000000
- 0.952543913321
- 0.000000000000000

PROBLEM 2 – UNIT CONVERSION

- Problem

Unit conversion

- idea

$x * A = y * B$

Already know x0 and y0

$x0 * A = y0 * B$

$A / B = y0 / x0$ and $B / A = x0 / y0$

$x = (B / A) * y = (x0 / y0) * y$

$y = (A / B) * x = (y0 / x0) * x$

Once we get x or y, we can solve the other variable

PROBLEM 3 -- CHEAP TRAVEL Codeforces Round #266 (Div. 2)

- Ann has recently started commuting by subway. We know that a one ride subway ticket costs a rubles. Besides, Ann found out that she can buy a special ticket for m rides (she can buy it several times). It costs b rubles. Ann did the math; she will need to use subway n times. Help Ann, tell her what is the minimum sum of money she will have to spend to make n rides?
- **Input**
- The single line contains four space-separated integers n, m, a, b ($1 \leq n, m, a, b \leq 1000$) — the number of rides Ann has planned, the number of rides covered by the m ride ticket, the price of a one ride ticket and the price of an m ride ticket.
- **Output**
- Print a single integer — the minimum sum in rubles that Ann will need to spend.

PROBLEM 3 – CHEAP TRAVEL

- Problem:

Ann has two choices. First one is buying one ticket using A rubes. Second one is buying m tickets using B rubes. Now she has n rides. How much rubes does she need to pay?

- idea:

Ann always wants to spend the least money to get n rides. We can first calculate the cost for every ticket if she use B rubes to buy m tickets.

C2 = b / m.

Then we compare it with C1 which is A.

If C1 <= C2. It means that she should buy every ticket one by one. (n*A)

If C2 < C1. It means that she could buy as many as tickets by buying m tickets at a time and then buy the rest tickets one buy one or should could buy all tickets, or she could all tickets by buying m tickets at a time .

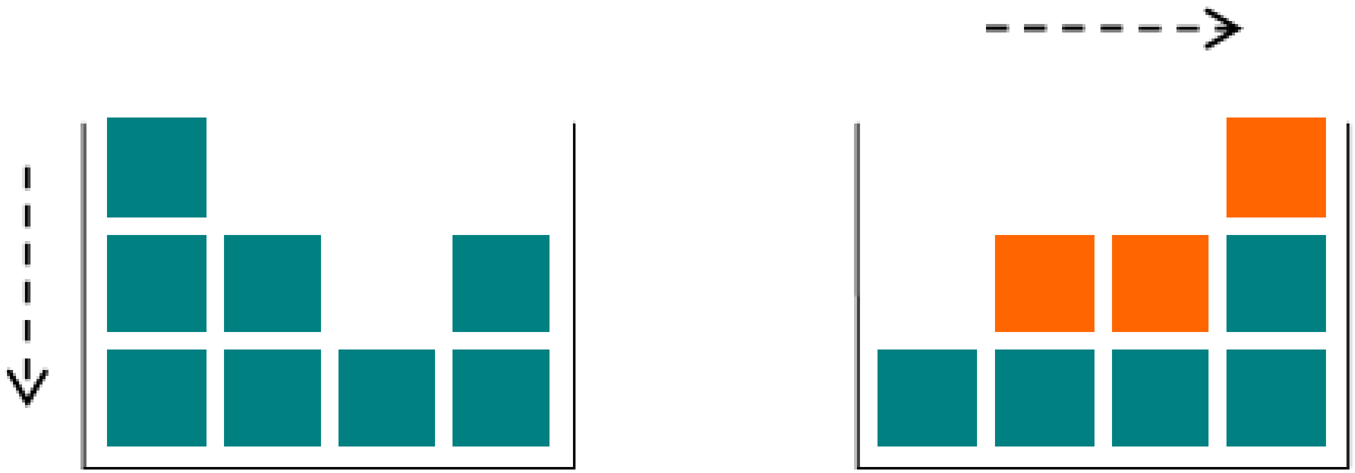
(min([n / m] * B + (n mod m) * A, ([n / m] + 1) * B).

PROBLEM 4 – GRAVITY FLIP Codeforces Round #238 (Div. 2)

- Little Chris is bored during his physics lessons (too easy), so he has built a toy box to keep himself occupied. The box is special, since it has the ability to change gravity.

PROBLEM 4 – GRAVITY FLIP

- There are n columns of toy cubes in the box arranged in a line. The i-th column contains a_i cubes. At first, the gravity in the box is pulling the cubes downwards. When Chris switches the gravity, it begins to pull all the cubes to the right side of the box. The figure shows the initial and final configurations of the cubes in the box: the cubes that have changed their position are highlighted with orange.



PROBLEM 4 – GRAVITY FLIP

- Given the initial configuration of the toy cubes in the box, find the amounts of cubes in each of the *n* columns after the gravity switch!
- **Input**
- The first line of input contains an integer *n* ($1 \leq n \leq 100$), the number of the columns in the box. The next line contains *n* space-separated integer numbers. The *i*-th number a_i ($1 \leq a_i \leq 100$) denotes the number of cubes in the *i*-th column.
- **Output**
- Output *n* integer numbers separated by spaces, where the *i*-th number is the amount of cubes in the *i*-th column after the gravity switch.

PROBLEM 4 – GRAVITY FLIP

Input:

```
3 2 1 4
```

Output:

```
1 2 2 3
```

Input:

```
3
2 3 8
```

Output:

```
2 3 8
```

PROBLEM 4 – GRAVITY FLIP

- Change the columns into rows. Every time when we add one column into the box we count the number of blocks in each row. Assume there're m rows.
- Re-mark all the columns. Make the rightmost column column #1, to the left, column #2, column #3, column #4 …… column #n
- Calculate the number of blocks in each column after the change of gravity. For each column, search for all the rows:
    - If (row(j)) <= # of column, num(col) increase by 1
- Output, print the num(col) inversely, means from column #n to column #1.

PROBLEM 5 – MEGACITY

Codeforces Round #242 (Div. 2)

The administration of the Tomsk Region firmly believes that it's time to become a megacity (that is, get population of one million). Instead of improving the demographic situation, they decided to achieve its goal by expanding the boundaries of the city.

The city of Tomsk can be represented as point on the plane with coordinates (0; 0). The city is surrounded with n other locations, the i-th one has coordinates (x_i, y_i) with the population of k_i people. You can widen the city boundaries to a circle of radius r. In such case all locations inside the circle and on its border are included into the city.

Your goal is to write a program that will determine the minimum radius r, to which is necessary to expand the boundaries of Tomsk, so that it becomes a megacity.

PROBLEM 5 – MEGACITY

Input
The first line of the input contains two integers n and s ($1 \leq n \leq 10^3$; $1 \leq s < 10^6$) — the number of locatons around Tomsk city and the population of the city. Then n lines follow. The i-th line contains three integers — the x_i and y_i coordinate values of the i-th location and the number k_i of people in it ($1 \leq k_i < 10^6$). Each coordinate is an integer and doesn't exceed 10^4 in its absolute value.
It is guaranteed that no two locations are at the same point and no location is at point (0; 0).

Output
In the output, print "-1" (without the quotes), if Tomsk won't be able to become a megacity.
Otherwise, in the first line print a single real number — the minimum radius of the circle that the city needs to expand to in order to become a megacity.
The answer is considered correct if the absolute or relative error don't exceed 10^{-6}.

PROBLEM 5 – MEGACITY

- Idea

1. Sort the radiuses of the points by formula r = sqrt((x – 0) ^ 2 + (y – 0) ^ 2).
2. Go from the most inside point to outside points and add the k values. (the radiuses are strictly increasing).
3. If at some point the total k value exceeds 1 million, then the radius of the point is the smallest radius. Print the radius.
4. If the total k value of all the points is smaller than 1 million. Print "-1".

PRACTICE RESOURCES

A2 judge

https://a2oj.com/category?ID=92