

COMPUTER SCIENCE CLUB

Intro to Competitive Programming (Pt. 4)

Praneeth Suryadevara



Implementation/Simple Math Problems

- Most problems on the CCC (especially J1-J5, S1-S2)

❌	CCC '17 J1 - Quadrant Selection	CCC	Implementation	3
✅	CCC '17 J2 - Shifty Sum	CCC	Simple Math	3
✅	CCC '17 J3 - Exactly Electrical	CCC	Simple Math	3p
✅	CCC '17 J4 - Favourite Times	CCC	Simple Math	5p
❌	CCC '17 S1 - Sum Game	CCC	Implementation	3
✅	CCC '17 S2 - High Tide, Low Tide	CCC	Implementation	5
✅	CCC '17 S3 - Nailed It!	CCC	Simple Math	7p
❌	CCC '17 S4 - Minimum Cost Flow	CCC	Graph Theory	17p
❌	CCC '17 S5 - RMT	CCC	Data Structures	20p

- Important to master before we proceed to complex topics like graph theory and dynamic programming

Strategy

- Most of these problems have solutions that can be first expressed in words/thoughts/diagrams, then implemented
- Can try a simple algorithm in your head/on paper and match it with the example inputs/outputs
- Discovering the correct strategy is 50% of the work: the other 50% is implementing it

Example Problem: J3 - Art

Mahima has been experimenting with a new style of art. She stands in front of a canvas and, using her brush, flicks drops of paint onto the canvas. When she thinks she has created a masterpiece, she uses her 3D printer to print a frame to surround the canvas.

Your job is to help Mahima by determining the coordinates of the smallest possible rectangular frame such that each drop of paint lies inside the frame. Points on the frame are not considered inside the frame.

Identify the objective:

- Determine the coordinates of the smallest rectangle that fits all paint drops

Input/Output Specification

Input Specification

The first line of input contains the number of drops of paint, N , where $2 \leq N \leq 100$ and N is an integer. Each of the next N lines contain exactly two positive integers X and Y separated by one comma (no spaces). Each of these pairs of integers represent the coordinates of a drop of paint on the canvas. Assume that $X \leq 100$ and $Y \leq 100$, and that there will be at least two distinct points. The coordinates $(0, 0)$ represent the bottom-left corner of the canvas.

Output Specification

Output two lines. Each line must contain exactly two non-negative integers separated by a single comma (no spaces). The first line represents the coordinates of the bottom-left corner of the rectangular frame. The second line represents the coordinates of the top-right corner of the rectangular frame.

- First input: N - the amount of coordinates to be collected
- Next N lines - X, Y coordinates for drops of points
- Output: Create the smallest possible canvas to fit all points
 - Output the bottom-left and top-right corner of the rectangle

Input/Output Example

Sample Input

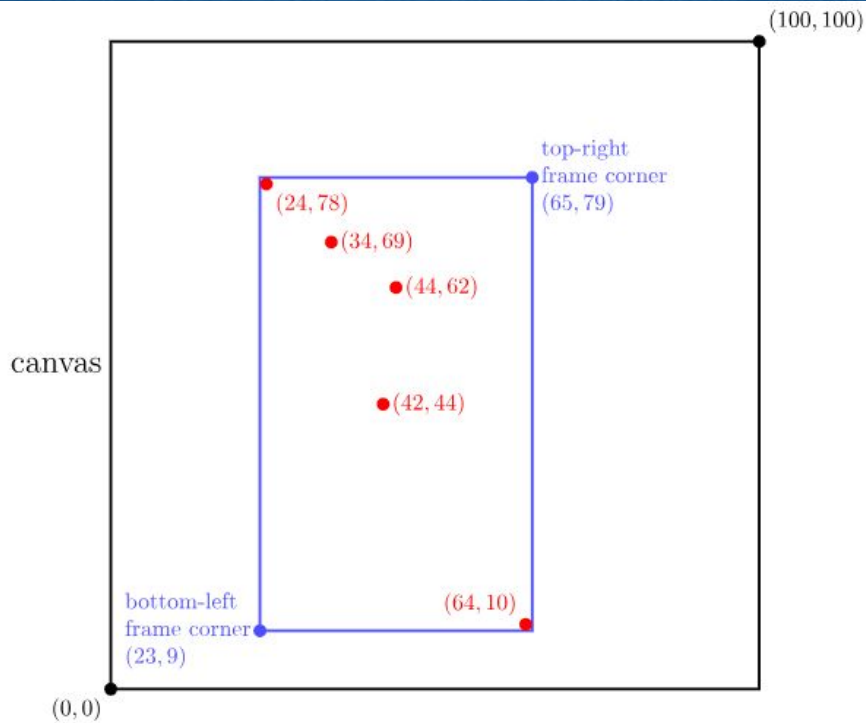
5
44,62
34,69
24,78
42,44
64,10

Output for Sample Input

23,9
65,79

- Based on this, what would be your first approach to solving this problem?
 - Take 2 minutes to brainstorm: Look at patterns in the input/output

Solution



Strategy:

- The corners of the frame are one greater/less than the highest/lowest x and y coordinates
- Any more would be wasting space; any less wouldn't fit all points

Code Solution

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main()
5  {
6      // Start the max at 0 and min at 102; make sure the initial values are not final
7      int xmax = 0; int ymax = 0; int xmin = 102; int ymin = 102;
8      int points; cin >> points;
9
10     for (int i = 0; i < points; i++) { // Iterate through each point
11
12         int xVal; int yVal; char comma; // initialize and input variables
13         cin >> xVal >> comma >> yVal;
14
15         // If the values of the point exceed the current min/max, change it to that value
16         if (xVal > xmax) {xmax = xVal;}
17         if (xVal < xmin) {xmin = xVal;}
18         if (yVal > ymax) {ymax = yVal;}
19         if (yVal < ymin) {ymin = yVal;}
20     }
21     // Output one less than the collected minimums, one more than the collected maximums
22     cout << xmin-1 << "," << ymin-1 << endl;
23     cout << xmax+1 << "," << ymax+1 << endl;
24 }
```

- Setting the max value to a value lower than the range, and a min value to something higher than the range
- Makes sure the initial values aren't the final values from the collected data

Problems to Solve

Beginner:

- Exactly Electrical (2017) - J3/3p

Intermediate:

- Crazy Fencing (2021) - S1/5p

Advanced:

- Geneva Confection (2014) - S3/7p
- Nailed It (2017) - S3/7p

Questions 10p+ fall into other categories

Find these problems on DMOJ by searching the title or looking for the year and name under the CCC category

Code Solutions: Exactly Electrical

```
1  #include <bits/stdc++.h>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      int x1, y1, x2, y2, moves;
7      cin >> x1 >> y1 >> x2 >> y2 >> moves;
8
9      moves -= (abs(x1-x2) + (abs(y1-y2)));
10     if (moves % 2 == 0 && moves >= 0) {
11         cout << "Y";
12     }
13     else {
14         cout << "N";
15     }
16     return 0;
17 }
```

Strategy:

- The net path is the difference between the initial and final x and y coordinates
- Any other movements need to be offset by a move in the other direction
- Hence, if the remaining moves is divisible by 2, it's doable

Solutions: Any other Problems?

- Can solve any of the other problems on DMOJ live
- Any other questions? Can email me (708517@pdsb.net) if you have any questions about the three other problems linked in the slideshow