**<https://www.iarcs.org.in/inoi/2004/inoi2004/inoi2004-solutions.php>**

**INOI 2004, Solutions**

[**IARCS home**](https://www.iarcs.org.in/)**>**[**OLYMPIAD**](https://www.iarcs.org.in/inoi)**>**[**Archives**](https://www.iarcs.org.in/inoi/archives.php)

**Question 1**

Finding whether a path exists is straightforward. Start from the top left corner and, for each square, recursively explore all neighbours that are at most one unit higher or lower than the current square. A path exists provided this recursive exploration reaches the bottom right corner.

To identify a path, enhance the recursive search by recording the direction from which you came. For example, if your search takes you one square down, store the character 'u' at the new position to say that you reached this square from the *up* direction. In the end, you can follow the directions back from the bottom right square to reach the top left square.

Unfortunately, this provides the path that we wish to display in *reverse*. Rather than reversing the path, we can reverse the direction of exploration! Starting from the bottom right corner, we recursively explore all neighbours and see if we can reach the top left corner. Now, the directions that we accumulate along the way provide a path from the top left corner to the bottom right corner

* [C program for this solution](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr.c).
* [C program to check that a solution is valid](https://www.iarcs.org.in/inoi/2004/inoi2004/tahrcheck.c).  
    
  This program takes an input file name and output file name as command line arguments and verifies if the solution recorded in the output file is valid for this input. For instance:
* tahrcheck 1.in 1.out

checks that the data in 1.out correctly solves the input instance 1.in

* Test inputs and outputs:
  + 4x4 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-1.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-1.out).
  + 30x30 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-2.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-2.out).
  + 50x50 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-3.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-3.out).
  + 60x60 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-4.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-4.out).
  + 100x100 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-5.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-5.out).
  + 200x200 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-6.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-6.out).
  + 300x300 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-7.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-7.out).
  + 600x600 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-8.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-8.out).
  + 1000x1000 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-9.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-9.out).
  + 1000x1000 grid: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-10.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-10.out).
  + [All test cases, zipped](https://www.iarcs.org.in/inoi/2004/inoi2004/tahr-data.zip).

**Question 2**

From the problem statement, it is clear that wrestlers who win more matches should be ahead in the queue to meet the Rajah. Thus, for *N*wrestlers, we can compute the outcome of all *N(N-1)* matches and record the number of matches won by each wrestler. If we sort this list in descending order based on the number of wins, we are done.

What if a group of wrestlers have the same number of wins? How should we arrange such a group in the queue? A little analysis shows that this cannot happen. If wrestler *A* beats wrestler *B* and wrestler *B* beats wrestler *C*, then *A* also beats *C*!

To see this, let *sA*, *sB* and *sC* denote the strengths of *A*, *B* and *C* and let *rA*, *rB* and *rC* denote the power of their rings, respectively. If *A* beat *B*, we know that *sA + rA\*sB > sB + rB\*sA*. If we collect the *sA* terms on the left and the *sB* terms on the right, we have *sA(1-rB) > sB(1-rA)*, or *sA/(1-rA) > sB/(1-rB)*.

Thus if *A* beats *B* and *B* beats *C*, we have *sA/(1-rA) > sB/(1-rB)* and *sB/(1-rB) > sC/(1-rC)*. It is immediate that *sA/(1-rA) > sC/(1-rC)*, so *A* beats *C*.

From this it follows that we can directly sort the wrestlers in descending order by the value *s/(1-r)*, without computing the outcomes of all *N(N-1)*matches.

* [C program for this solution that uses bubble sort](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-bsort.c).
* [C program for this solution that uses quicksort](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-qsort.c).
* Test inputs and outputs:
  + 10 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-1.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-1.out).
  + 20 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-2.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-2.out).
  + 50 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-3.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-3.out).
  + 100 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-4.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-3.out).
  + 100 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-5.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-4.out).
  + 200 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-6.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-5.out).
  + 500 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-7.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-5.out).
  + 1000 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-8.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-6.out).
  + 5000 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-9.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-6.out).
  + 10000 wrestlers: [Input](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-10.in), [Output](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-6.out).
  + [All test cases, zipped](https://www.iarcs.org.in/inoi/2004/inoi2004/wrestler-data.zip).