

PROGRAMMING CHALLENGE

The engineering team at Churchill Navigation is challenging you to beat our solution.

In a nutshell: We give you a bunch of ranked 2D points, then ask you to find the most important ones inside some randomly generated rectangles. Easy, right? Now make it fast!

To get started, just download [this .zip file](#). It contains the problem/interface specification, the testing framework and the reference solution. Note: the reference solution is only provided to check your results. Beating the reference solution doesn't mean you have a fast solution. You want to smash it! The fastest solutions are consistently over 1000 times faster than the reference.

For Applicants

Solving the challenge requires building a DLL – something many people have never done before. Figuring this out is the first step. It shows us that you can learn independently and can use the tools of the trade.

The second component of the challenge is algorithmic. A fast solution will require using a data structure and writing a traversal algorithm. The very fastest solutions may require a unique data structure designed specifically for this problem. Designing a good data structure and algorithm shows us that you have a strong understanding of the fundamentals of computer science.

The last part of the challenge is optimization. It's often possible to improve the performance of your algorithm by a factor of 10 or more by the clever use of optimizations. This component shows a deep understanding of the compiler and the way the processor actually runs your code. Optimization is not necessary for a fast solution, but it can often separate a good solution from a truly great one.

For Challengers

Pit your algorithm against the best. Get to the top. The challenge starts with an equal playing field. May the best code win!

The Rules Are Simple

You can submit a solution multiple times.

All submissions will become public-domain DLLs.
Churchill Navigation independent contractors and employees are not eligible to participate.

Solution Qualifications

You must provide the DLL (including all non-OS dependencies, if not statically linked), source code and project files.
The DLL must load successfully and not crash during testing with the provided point_search.exe test framework.
All solutions will be tested on an Intel NUC computer (model D54250WYKH) with Intel i5-4250U CPU and 16GB RAM, running Windows 8.1 Pro 64 bit.
Maximum memory commit size: 512MB (the actual committed memory is reported by the test framework)
Maximum loading time: 30 seconds (the actual load time is reported by the test framework)
Maximum search time: 10 seconds (the actual search time is reported by the test framework)
The results must match with the provided reference.dll (comparison is done by the test framework, when also specifying reference.dll as a module)
The score for your solution is determined solely by the search time.
The points and search rectangles are randomly generated. For the sake of reproducibility, we will run the tests with a fixed random seed.
Cheating (ie. various ways of tricking the test framework to report false results) will also result in disqualification.

No Messing with the Code

To guarantee no modifications to our solutions, SHA-1 hashes are published for all our binaries.
point_search.exe (the test framework): 1f3b357e99bb14f37638806e42cb16dc8e5a69c8
reference.dll (slow reference solution used for correctness testing) : 3f913856b4f008877fcc0fce5dd7d1d364cda730
churchill.dll (our fast solution, not yet published): 1a3d656af3334629ec114c6ba59e1bf7460421bd

Submission of Your Solution

Send your solution (DLL, source code and project files) to challenge@churchillnavigation.com .
To avoid our mail server rejecting attachments with DLLs, we recommend packaging up all the files in an archive with file name encryption enabled.
The name of your DLL will be used on the public scoreboard, so name it something unique.

LET'S TALK

-  | Support: +1 720-744-3300
-  | Sales: +1 720-943-4ARS (4277)
-  | info@churchillnavigation.com