Chain game

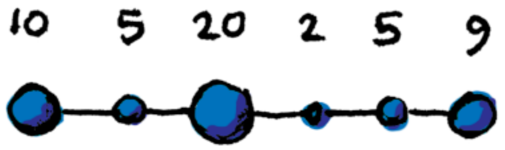
Starts: 13 Feb, 12:00 pm UTC

(Ends 20 Feb, 12:00 pm UTC)

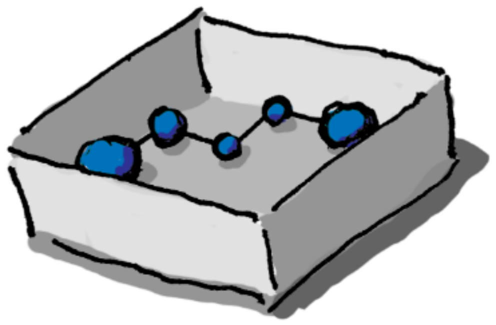
Our contest, the Chain Game, involves optimal two-dimensional folding. In this contest, you are given a chain and a box. Each link in the chain has a different weight, and your job will be to place the chain in the box as compactly as possible. You will be writing a function in MATLAB to do this.

The Chain Game

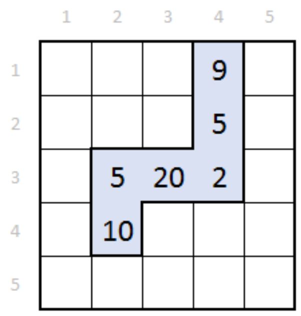
In this contest, you are given a chain and a box. Each link in the chain is exactly one unit long, but each one has a different weight. It looks something like this.



Your job is to place the chain in the box as compactly as possible. The constraint is that the chain must lie in a single plane and snake through the box using only straight segments or 90 degree turns.

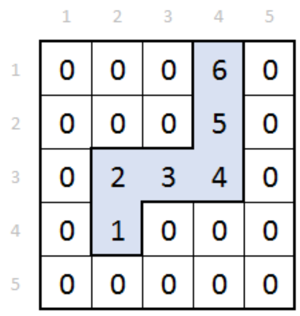


So for example, if you are given the chain vector c = [10 5 20 2 5 9] you might choose to put it into a 5-by-5 box like so.



$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & & & & 9 & \\ 2 & & & & 5 & \\ 3 & & 5 & 20 & 2 & \\ 4 & & 10 & & & \\ 5 & & & & & \end{array}$$

If this is how you want to arrange the chain, your code would return a box matrix b like the one shown below. The first element of the chain, with weight 10, goes into row 4, column 2. The second element, with weight 5, goes just above it at location b(3,2), and so on.

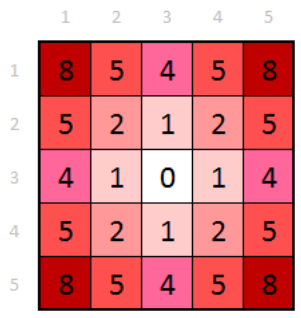


$$\begin{array}{c|ccccc} & 1 & 2 & 3 & 4 & 5 \\ \hline 1 & 0 & 0 & 0 & 6 & 0 \\ 2 & 0 & 0 & 0 & 5 & 0 \\ 3 & 0 & 2 & 3 & 4 & 0 \\ 4 & 0 & 1 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 \end{array}$$

We will measure compactness by calculating the rotational moment of inertia (https://en.wikipedia.org/wiki/Moment_of_inertia) around the center of the matrix. Physically you can imagine that we are putting the box on a turntable and spinning it. You want the box to spin with as little effort as possible. Mathematically, the moment of inertia is calculated by multiplying the mass of each link by the square of the distance to the center of the box matrix.

$$I_{rot} = \sum_n m_n r_n^2$$

In our simplified matrix problem, this means each position in the box has a penalty based on how far it is from the center. For a 5-by-5 box, this distance penalty matrix p looks like this.



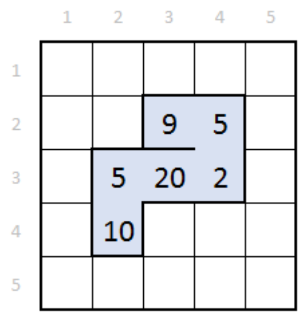
	1	2	3	4	5
1	8	5	4	5	8
2	5	2	1	2	5
3	4	1	0	1	4
4	5	2	1	2	5
5	8	5	4	5	8

The box is always a square matrix with an odd number side length n. So there is zero penalty for whatever link occupies the center of the box. That makes it a good place to put your heaviest link, which is exactly what we've done here. Our result is thus

```
res = 10*2 + 5*1 + 20*0 + 2*1 + 5*2 + 9*5
```

```
res = 82
```

We pay a high price for the 9 in the first row of the box. Let's twist the 9 down to location b(2,3) as shown here.



```
res = 10*2 + 5*1 + 20*0 + 2*1 + 5*2 + 9*1
```

```
res = 46
```

A significant improvement!

The Syntax

You must write a function that has this signature.

```
function b = chain(c,n)
    % Define b here...
    b = zeros(n);
end
```

The n-by-n matrix b must have the sequence c snaking through it. Thus each number 1..length(c) must appear exactly once and in adjacent elements of b. All other positions must contain zeros.

You can download a sample test suite here (http://cumuluscontest.org/download_file/view/4/182), but bear in mind we will be using a different test suite to determine the final scores.

Scoring

The overall score for your entry is a combination of two things:

- Your average score across all the game boards (result)
- How fast your code runs (runtime)

These three factors are passed to our scoring algorithm to produce a final score, according to the equation:

$$score = k_1 * result + k_2 * e^{(k_3 * runtime)}$$

Both of these are to be minimized and the lowest overall score at the end of the contest wins. We don't publish the values k1-k3, but they aren't hard to figure out.

About the Contest

Once an entry has been submitted, it cannot be changed. However, any entry can be viewed, edited, and resubmitted as a new entry. You are free to view and copy any entry in the queue. If your modification of an existing entry improves its score, then you are the "author" for the purpose of determining the winners of this contest. We encourage you to examine and optimize existing entries.

Starting and ending times are based on noon in St Andrews, UK, but the web pages will show all times in Coordinated Universal Time (UTC).

Prizes

We'll be awarding a special St Andrews merchandise bundle for the Grand Prize winner, but we'll also have random MATLAB shirt give-aways throughout the contest, organised as follows:

- *most active* - all players who submit at least one entry on each of the 7 days of the contest
- *random daily win* - each day at 12 p.m. we will randomly pick a player who has submitted at least one entry that day
- *random daily leader* - each day at 12 p.m. we will randomly pick a time and award the top entry submitted closest to that time

Fine Print

The allowable functions are those contained in the basic MATLAB package available in $MATLAB/toolbox/matlab, where $MATLAB is the root MATLAB directory. Functions from other toolboxes will not be available. Entries will be tested against the MATLAB.

The following are prohibited:

- MEX-files
- Java commands or object creation
- eval, feval, inline, function handles, etc.
- Shell escape such as !, dos, unix
- Handle Graphics commands
- ActiveX commands
- File I/O commands
- Debugging commands
- Printing commands
- Simulink commands
- Benchmark commands such as tic, toc, flops, clock, pause
- error, clear, persistent

Hacking

Your entry will time out and be disqualified if it takes more than 50 seconds. You can submit as many entries as you like, but entries that compromise the contest machinery are not allowed. Out of consideration for everyone participating in the contest, we ask that you not abuse the system.

Extraction of puzzles in the test suite by manipulating the score, runtime, or error conditions is also forbidden. Tuning the entry to the contest test suite via multiple entries is permitted, but we ask that you not overwhelm the queue.

We'll be monitoring the queue - if we find signs of malicious behaviour we'll be issuing warning, which could lead to temporary or permanent bans.

This is a research project run by the University of St Andrews. If you have any questions or concerns, please contact Elena Miu (em95@st-andrews.ac.uk (mailto:mailto:em95@st-andrews.ac.uk)) or Dr. Luke Rendell (ler4@st-andrews.ac.uk (mailto:mailto:ler4@st-andrews.ac.uk))

Logged in as [illegible] Logout (http://cumuluscontest.org/login/logout/1487575271%3Afb77c4db59ef15f41a8cfa93e78a13ef)

Sponsored by MathWorks®