

From Euler to Stellaris: New Variants of Random Walks

David Aldous

24 February 2023

Hi Chat-GPT

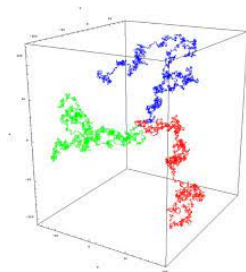
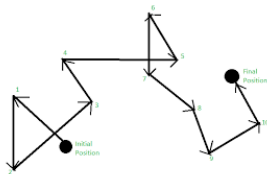
In mathematics, what is a random walk?

In mathematics, a random walk is a process that describes a sequence of steps or movements, where each step is determined randomly. The concept is used to model the behavior of a variety of systems, including stock prices, molecular movement, and the spread of diseases.

In a random walk, each step can be thought of as a change in the position of an object, where the magnitude and direction of the change are determined randomly. The movement is typically described in terms of time, so that each step in the sequence occurs at a specific time, and the position of the object changes with each step. The position of the object after n steps is the sum of the positions of the object after each of the n individual steps.

Random walks are widely used in a variety of fields, including mathematics, physics, finance, and computer science, and have applications in areas such as risk analysis, optimization, and machine learning.

Chat-GPT is talking about random walk in d dimensional space, formally $S_n = \sum_{i=1}^n X_i$ for i.i.d. X_i .



Slightly different is random walk on a graph.

Hi Chat-GPT

In mathematics, what is a random walk on a graph?

A random walk on a graph is a type of random walk where the steps are taken along the edges of a graph, rather than in a continuous space.

In a random walk on a graph, at each step, the walker randomly selects one of the edges emanating from the current vertex and moves to the vertex at the other end of that edge. The direction of the walk is thus determined by the edge chosen, and the walker moves through the graph according to the structure of the graph.

Random walks on graphs are widely used in various fields, including computer science, physics, and mathematics, to model complex systems. For example, they can be used to model the behavior of particles moving through a network, the spread of information or disease in a network, and the behavior of search algorithms.

The behavior of random walks on graphs can be characterized by various statistical measures, such as the mean first passage time, which is the average time it takes for the walker to reach a certain vertex for the first time, and the stationary distribution, which describes the long-term behavior of the walker. These measures can be used to study the properties of the graph, such as its connectivity and centrality, and can provide insights into the behavior of the underlying system being modeled.

<https://i.stack.imgur.com/Hqiqs.gif>

If you are starting research in some math-related topic, then you probably feel that every specific question that can be posed in non-technical terms has already been studied by other people.

And that's usually correct. MathSciNet shows 7463 papers with *random walk* in their title.

However, this talk describes 3 variants of random walk that have not been studied (much).

1. Random Eulerian circuits

A rather obvious observation in introductory graph theory is

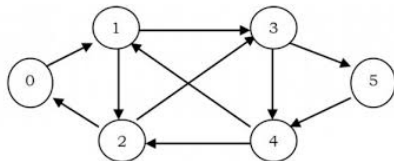
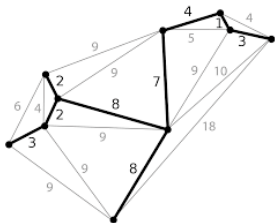
Lemma

*A finite connected undirected graph has at least one **spanning tree**, that is a connected edge-subgraph which is a tree.*

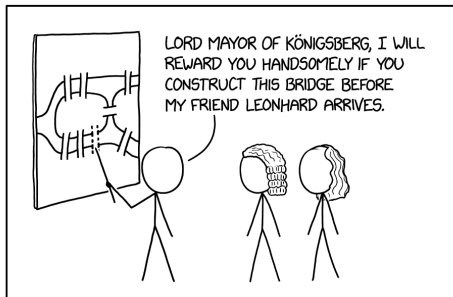
Leonhard Euler proved a version of “the first theorem in graph theory”.

Theorem

*A finite, strongly connected, directed graph which is balanced (each vertex has in-degree = out-degree) has at least one **Eulerian circuit**, that is a tour using each directed edge exactly once.*



KÖNIGSBERG



I TRIED TO USE A TIME MACHINE TO CHEAT ON MY ALGORITHMS FINAL BY PREVENTING GRAPH THEORY FROM BEING INVENTED.



PERMANENT LINK TO THIS COMIC: [HTTPS://XKCD.COM/2694/](https://xkcd.com/2694/)

IMAGE URL (FOR HOTLINKING/EMBEDDING): [HTTPS://IMGS.XKCD.COM/COMICS/KONIGSBERG.PNG](https://imgs.xkcd.com/comics/konigsberg.png)

Within probability theory, there is a large literature on **uniform random** spanning trees, because they relate to many other discrete structures – see Lyons - Peres monograph *Probability on Trees and Networks*.

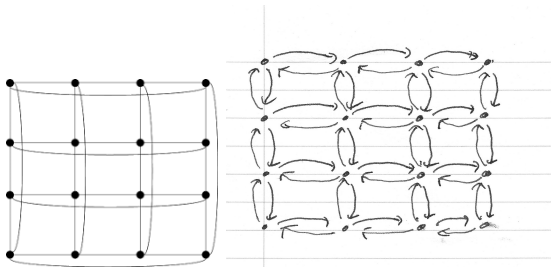
In contrast, there is very little literature (2 papers?) on uniform random Eulerian circuits. This is curious because there's a **surprising connection** between the two topics.

In a balanced directed graph, choose any spanning tree, with directed edges toward an arbitrary root. Then, from the root, do a walk, at each stage arbitrarily choosing an unused edge but saving the spanning-tree-edge until last. This always gives an Eulerian circuit [easy].

True (but **not obvious**) that with a *uniform* random spanning tree and *uniform* random walk-step choices we get a *uniform* random Eulerian circuit.

Fact: It is quite easy to simulate a uniform random spanning tree (of an arbitrary finite connected graph). So we can then simulate a uniform Eulerian circuit on a balanced graph (paper #1).

Now as a simple example let us consider the discrete torus \mathbb{Z}_N^d . Replace each edge by 2 directed edges. So in-degree = out-degree = $2d$. Any Eulerian circuit consists of $2d$ **excursions** from the origin.



We understand simple random walk on the infinite lattice \mathbb{Z}^d very well. In particular, for $d \geq 3$ simple random walk on \mathbb{Z}^d is **transient**, meaning it returns to the start only finitely often. The intuition that the random Eulerian tour is like a constrained random walk strongly suggests the following, which is supported by simulation (paper #2, with undergrad Jiangzhen Yu):

Open problem. Prove (in fixed $d \geq 3$) that of the $2d$ excursions at the origin, each has length $O(1)$ or $\Omega(N^d)$, not of intermediate order (as $N \rightarrow \infty$).

I have no idea how one might prove this – can't do theoretical analysis of algorithm output.

Take-away message. There is an unexpected connection between random Eulerian tours and random spanning trees. Known for a long time, but **apparently never exploited**.

2: The Nearest Unvisited Vertex walk on a Graph

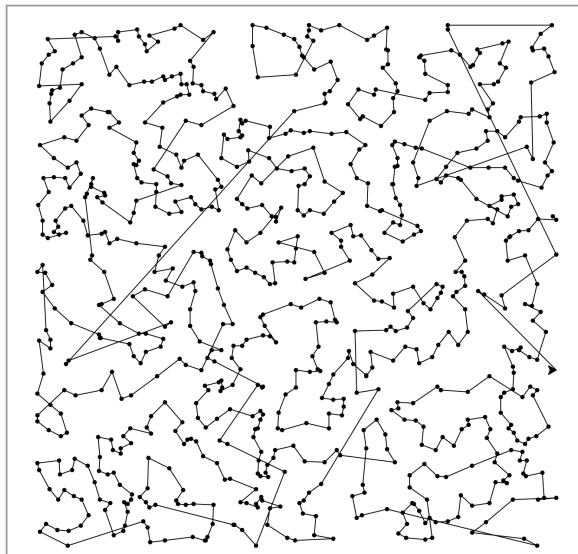
Consider a connected undirected graph G on n vertices, where the edges e have positive real lengths $\ell(e)$. Imagine a robot that can move at speed 1 along edges. We need a rule for how the robot chooses which edge to take after reaching a vertex. Most familiar is the “random walk” rule, choose edge e with probability proportional to $\ell(e)$ or $1/\ell(e)$. One well-studied aspect of the random walk is the **cover time**, the time until every vertex has been visited.

Instead of the usual random walk model, let us consider the *nearest unvisited vertex* (NUV) walk

after arriving at a vertex, next move at speed 1 along the path to the closest unvisited vertex

and continue until every vertex has been visited. Note this is deterministic and has some length (= time) $L_{NUV}(G, v_0)$ where v_0 is the initial vertex.

NUV walk on 800 random points in the square.
Simulation by undergrad Yechen Wang.



Of course *distance* $d(v, v')$ is shortest path length. In informal discussion we imagine lengths are scaled so that distance to closest neighbor is order 1, so L_{NUV} must be at least order n .

Natural first question: when is it $O(n)$ rather than larger order?

There is scattered old “algorithms” literature discussing the NUV walk as heuristics for TSP or as an algorithm for a robot exploring an unknown environment, but that literature quickly moved on to better algorithms.

I will say some results from my paper *The Nearest Unvisited Vertex Walk on Random Graphs*. Part 3 will explain one motivation.

There is a key starting math observation – implicit but rather obscured in the old literature. For now, we stay with non-random graphs.

Fractal dimension via ball-covering.

Question; How many (overlapping) radius- r balls are needed to cover a large solid region – say a cube or ball – in \mathbb{R}^d ? – call this number $N(r)$.

Answer: order V/r^d , where V is the volume to be covered. In other words

$$N(r) \asymp V/r^d \text{ for } 0 < r \ll \Delta := \text{diameter of region.}$$

We can apply the same “fractal dimension” idea to a large graph G with n vertices. Define $N(r) :=$ minimum number of centers s_1, s_2, \dots such that the union over s_i of $\text{Ball}(s_i, r)$ is the entire graph. If the graph G is such that we have an approximation

$$N(r) \approx nr^{-\dim}, \quad 1 \ll r \ll \Delta$$

then we can regard \dim as a *dimension* for the graph G .

Proposition (implicit in 1970s)

On any finite connected graph G_n with edge-lengths:

(i) $N(r) \leq 1 + L_{NUV}/r$, $0 < r < \infty$.

(ii) $L_{NUV} \leq 2 \int_0^{\Delta/2} N(r) dr$ where $\Delta = \max_{v,w} d(v,w)$ is the diameter of the graph.

So if G has a *dimension* in the sense

$$N(r) \approx nr^{-\dim}, \quad 1 \ll r \ll \Delta$$

then the Proposition has the informal interpretation that L_{NUV} is always $O(n)$ when $\dim > 1$.

Isolating that Proposition as the starting point, we can easily recover the two classical (1970s) results for non-random graphs.

Corollary

There is a constant A such that, for the complete graph on n arbitrary points in the area- n square, with Euclidean lengths,

$$L_{NUV} \leq An.$$

Note this implies the well known corresponding result $L_{TSP} \leq An$.

Corollary

Let $a(n)$ be the maximum, over all connected n -vertex graphs with edge lengths and all initial vertices, of the ratio L_{NUV}/L_{TSP} . Then $a(n) = \Theta(\log n)$.

The ball-covering relation is not helpful from the algorithms viewpoint. But it is useful for some **random** graph models. The simplest way to get a random graph with edge-lengths is to take a unweighted graph and then assign random edge-lengths. In that context, understanding the random “balls” around a center vertex is precisely the basic issue in **first passage percolation (FPP)**.

As the basic example, consider the random graph G_m that is the $m \times m$ grid, that is the subgraph of the Euclidean lattice \mathbb{Z}^2 , assigned i.i.d. edge-lengths $\ell(e) > 0$. with $\mathbb{E}\ell(e) < \infty$. Here $L_{NUV}(G_m)$ must be at least order m^2 . Using the shape theorem for FPP on \mathbb{Z}^2 one can show

Corollary

For the 2-dimensional grid model G_m above,

$$\mathbb{E}[L_{NUV}(G_m)] = O(m^2).$$

The same techniques would give $O(n)$ upper bounds in other simple models of n -vertex random graphs.

Open problems

- Are there general methods (subadditivity or local weak limits don't seem to work) to prove existence of a “law of large numbers” limit $c = \lim_n n^{-1} L_{NUV}(G_n)$ for simple random graph models?
- Evaluate c ?
- Order of magnitude of $\text{var}(L_{NUV})$ not clear from our small-scale simulations – seems $n^{1 \pm \epsilon}$. **Suitable for further undergrad research** – simulate the NUV walk on a range of different graphs.

Take-away message. There is an unexplored connection between the NUV walk on random graphs and FPP.

3: Games people play

I'm interested in probability and graphs; and also games.
Search MathSciNet for “graph and game” in title: get 654 papers.
None are games people actually play.

Are there “graph” games that millions of people do play?

Yes: Go, for instance.

But such traditional board games are closely tied to a fixed graph; I want games that can be played on a random graph, different every time you play. Are there any?

Well yes and no.

4X games

4X (abbreviation of Explore, Expand, Exploit, Exterminate) is a subgenre of strategy-based computer and board games, and include both turn-based and real-time strategy titles. The gameplay involves building an empire. Emphasis is placed upon economic and technological development, as well as a range of non-military routes to supremacy. (Wikipedia).

A representative game is *Stellaris*.

show <https://steamdb.info/app/281990/graphs/>

4X games are very complicated in detail. Much over-simplifying, let me invent a simple game which abstracts the common elements of the initial “Explore, Expand” phases, as follows.

My simple game. Copy the background setting of the NUV walk. There is a connected undirected graph G on n vertices, where the edges e have positive real lengths $\ell(e)$. You have a unit that you can move at speed 1 along edges. But you only see a neighborhood of the vertices that you have already visited. The “neighborhood” is defined so that you could (if you choose) implement the NUV walk. Make a game with k players, each with a unit moving simultaneously. A vertex you visit becomes part of your empire; other players cannot visit.

Easy fact: if at least one player is not completely stupid, this simple game will end with the vertices partitioned into the connected empires of the different players.

Goal: form the largest empire.

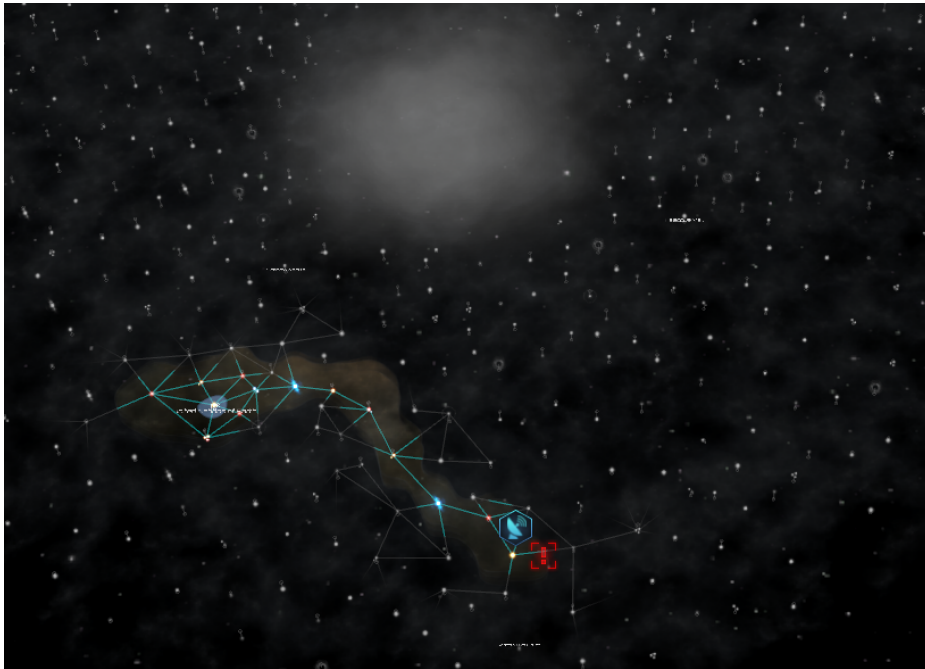
The graph is different every time you play, a realization of some unknown probability distribution on graphs.

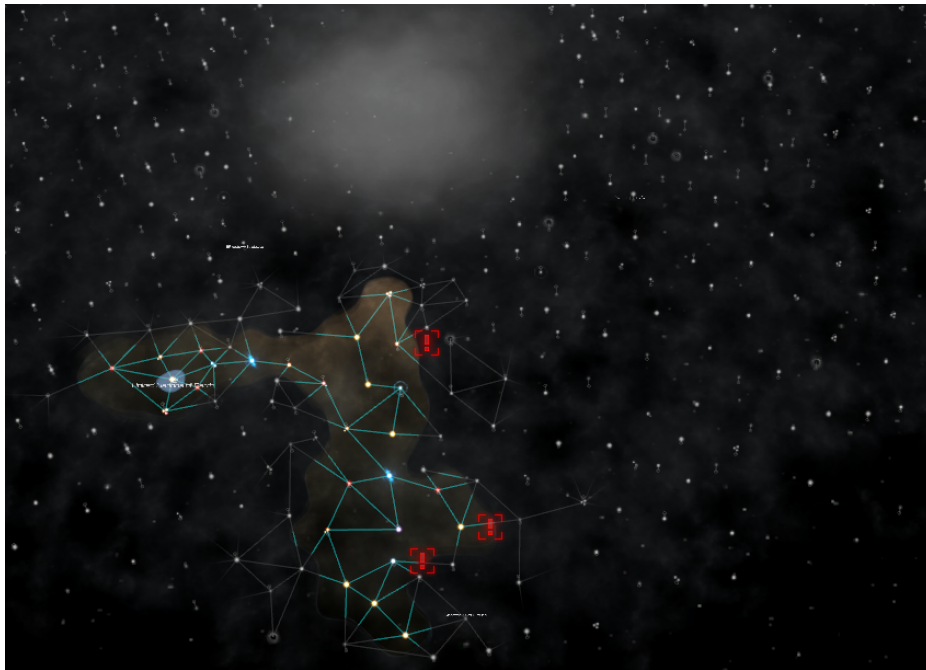
(very vague) Open Problem: What is a good strategy?

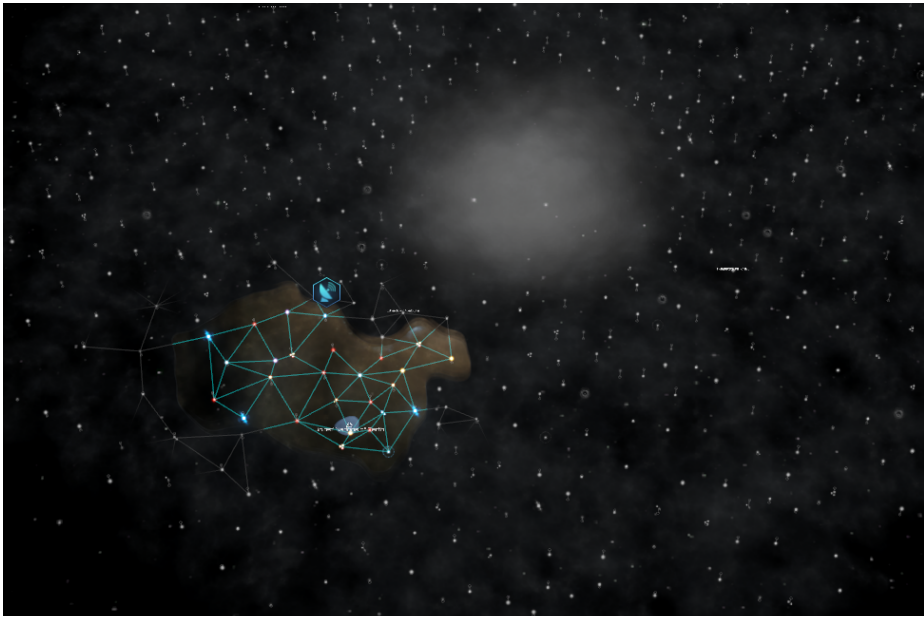
- **aggressive:** move away from starting vertex in some direction until meeting an opponent, then attempt to block.
- **defensive:** colonize a growing ball around your starting vertex.
- **NUV:** seems somewhat between.

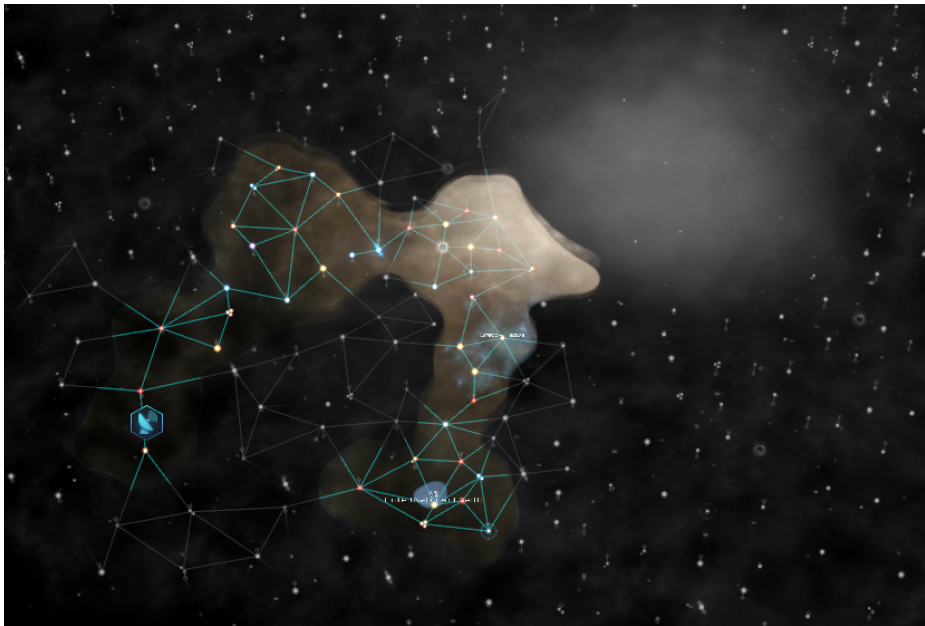
Intuitively, the best strategy depends on connectivity – for a locally tree-like graph with large visible neighborhood, “aggressive” is clearly better. Fun student project, in progress.

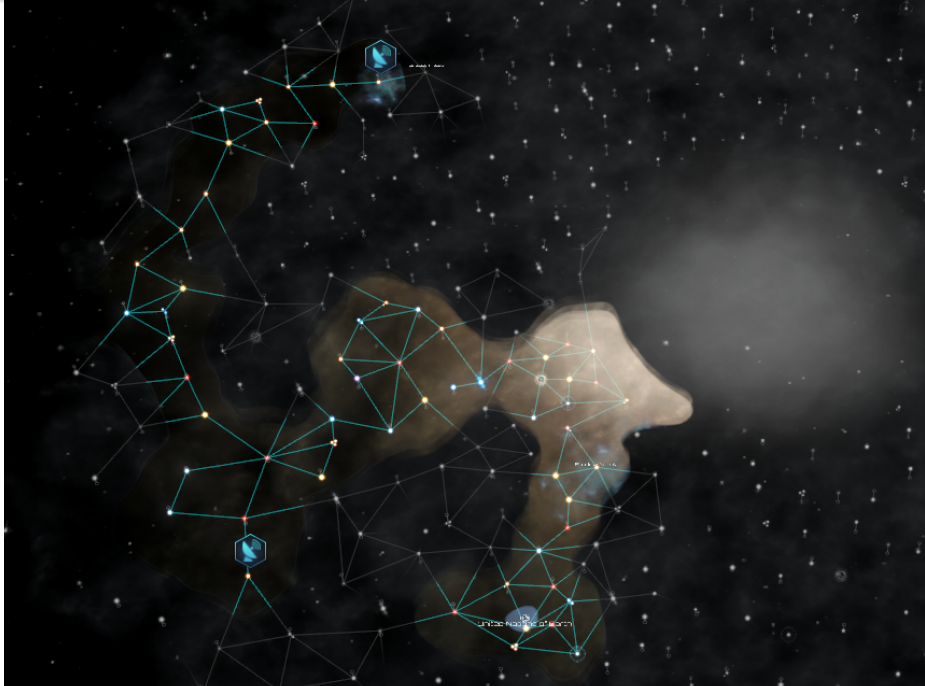
Take-away message? Clearly not do-able as theorem-proof mathematics, but good to “search away from the streetlight” and engage actual 21st century game activity.











Two undergrads (Peyton Repo and Daniel Kyutae Jung) worked on this project in Fall 2022.

A brief summary: it is hard to find a strategy that beats NUV. We are confident that a human would beat an opponent who used NUV, but we do not know an explicit strategy.

Future work:

(i) Explore other strategies. A naive implementation of the NUV strategy in the game has $O(n^3)$ time; needs some coding/algorithm skill to modify the Dijkstra algorithm as edges become unavailable.

(ii) Write code so that people could actually play this simple game online – crowdsource!