The Random Walk

Sena Ghobadi

November 1, 2023

Introduction

Imagine a drunkard taking a walk on the number line. The drunkard starts at 0 and takes steps of length 1 so that he may only end up on an integer after any given step. The man is inebriated enough that he possesses no sense of direction and will orient his steps indiscriminately. To be more quantitative, we assume that he may step to the left in the negative direction or to the right in the positive direction with equal probability ($p = 0.5$). For example, after $n = 1$ step, he may end up at $x_1 = -1$ or $x_1 = 1$. Though this thought experiment may seem arbitrary and contrived, it turns out that many natural and artificial phenomena are "drunk" in the sense that they obey this strange type of motion. Surprisingly, this characteristically random motion is ubiquitous and has uses stretching from modeling molecular motions to pricing financial derivatives[1]. This project will demonstrate a scenario built on this "random walk" concept. It will use the setup we just described.

Scenario

We return to the drunk man walking on the number line. Before we ask any particular question, let's better illustrate this motion by considering the possible outcomes after a few steps of our drunkard. As mentioned before, after 1 step, the man will end up in one of two positions, $x_1 = -1$ or $x_1 = 1$, with equal probability. For the second step, we can imagine performing the same process but for two different cases, one where our "origin" is now at -1 and another where our "origin" is at 1. For the $x_1 = -1$ case, the only two possible outcomes for the second step are traveling further left to $x_2 = -2$ or returning to $x_2 = 0$. Likewise, for the $x_1 = 1$ case, the drunkard may proceed to $x_2 = 2$ or return to $x_2 = 0$. After the second step, we note that there is only 1 way to get to $x_2 = -1$, 2 ways to get back to $x_2 = 0$, and only 1 way to get to $x_2 = -1$. We illustrate this "branching" type of counting in the table below up to $n = 3$ steps.

	-3	-2	-1	0	1	2	3
$n = 0$				1			
$n = 1$			1		1		
$n = 2$		1		2		1	
$n = 3$	1		3		3		1

Once we begin counting the possibilities after each step, the emerging pattern becomes quite obvious. The table enumerating the number of ways to reach a certain step also happens to match Pascal's Triangle which we know is an illustration of the binomial coefficients $\binom{n}{k}$. This suggests a connection to the binomial distribution that we've discussed in great depth throughout the semester. We establish this connection by assigning a step in the positive direction as a "success" and a step in the negative direction as a "failure". We define the number of successes to be k, and since the total number of steps is given by n, the number of failures must be $n - k$ since every step must result in either a success or failure. The total number of ways to obtain k successes in n steps is then the number of distinct ways to form a group of size k from n total elements i.e. $\binom{n}{k}$. Since there are only two possible direction for each step, the total number of possible "walks" that the drunkard may take is 2^n. We can then say that the probability of obtaining k successes is modeled as follows

$$P(k) = \frac{1}{2^n}\binom{n}{k}$$

Since k represents the number of steps to the right, the above distribution equivalently models the probability that the the drunkard will end up at position $k - (n - k) = 2k - n$ after n steps. For our first question, we ask the following:

[1]You can take a look at Brownian Motion or Wiener Process for more details.

First Question

Assume the drunkard has taken $2n$ steps (so that the number of steps is always even). What is the probability p_{2n} of returning to the origin ($x = 0$) on the final step? For this question, we don't care whether or not the drunkard has ever visited the origin before.

This question should be quite simple to answer given the result we just derived. Ending up at the origin after $2n$ steps corresponds to ending up at position $2k - 2n = 0$. Solving for k yields $k = n$, so the probability of ending up at the origin is

$$p_{2n} = P(n) = \frac{1}{2^{2n}}\binom{2n}{n}$$

What I find interesting about this result is that the probability seems to decay quite slowly (at least in how I intuit this process) for larger n. For instance, the probability of returning to the origin in $2n = 100$ (100 steps!) is still about 8%. An intuitive way of rationalizing this, is that there are many possible cases earlier on where the drunkard simply returns to the start. Once he's back at the start, it's as if any previous trials never occurred and he has made no "progress". So, appropriately enough for a drunkard, this random walk process is *memoryless.*

While this result is interesting, it very closely follows what we've already discussed in class. We offer a much more enriching problem by considering the question that inevitably follows from our previous result:

Second Question

What is the probability p_f of returning to the origin *for the first time* on the final step (we again take $2n$ total steps)?

This question is significantly more challenging to answer than the previous question since there isn't exactly a distinct distribution that we can point to solve our problem for us. The total number of "walk" possibilities remains as 2^{2n}, but counting the number of ways to end up at the origin without ever crossing it once during the entire journey is quite a challenge. One issue with counting all the possibilities, either directly or through complementary counting, is that there ends up being so many possible ways to return to the origin for larger n, so even if one experiments with small cases and attempts to induct, I suspect that they would either go insane searching for a pattern or end up with some ghastly product or series (this was my initial approach). I find that constructing some sort of spatial or geometric analogy with these types of problems often reveals some hidden structure that is obscured by the abstractions of mathematics. One disadvantage of imagining our whole problem on the number line is that we can't directly read off where the drunkard has been in the past. Thus, we introduce a new image to project our problem onto shown in the figure below (we illustrate the specific case of $n = 4$).

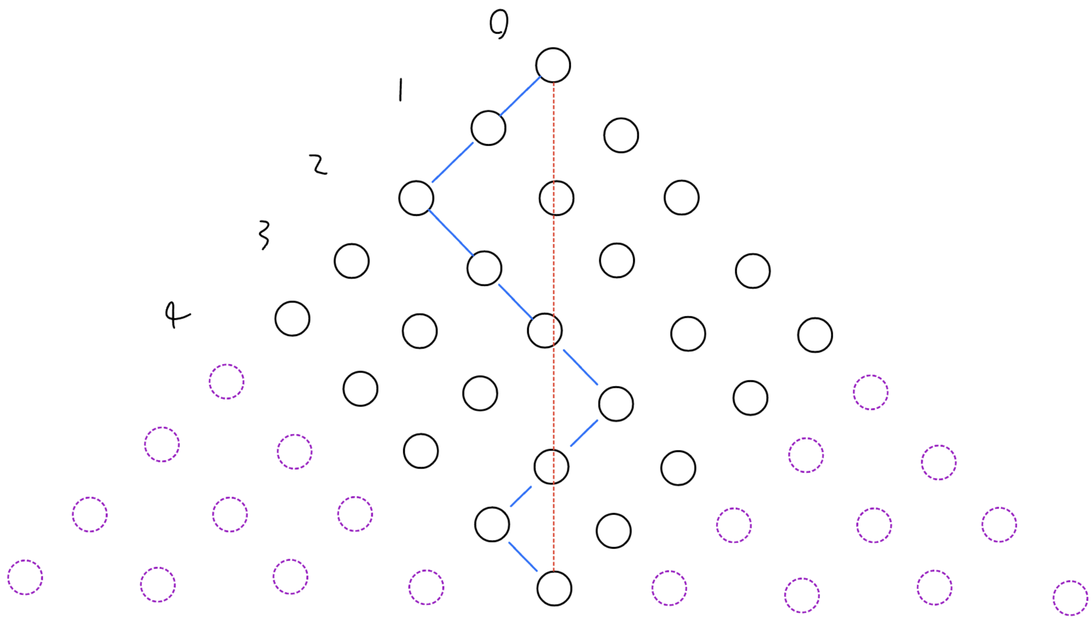

Figure 1: Illustration of "walk" on Pascal's triangle which allows us to observe the evolution of the walker in time.

Leveraging our intuition of Pascal's triangle, we assign a temporal dimension to our walker by depicting his walk *on top* of Pascal's triangle. The labels on the left designate the value of n the number of steps. Every bubble represents a possible position of the walker. The red line designates the position of the origin. Notably, if the walker crosses the red line (or simply touches it if it's not the final step), he has "failed" and violates the condition that he must make it back to the origin on the $2n$th step. The black bubbles are positions from which the walker can still make it back to the origin within the remaining steps. The purple bubbles represent positions from which the walker can never make it to the origin in the remaining steps. In our example above, we depict an example path with the blue line. This blue line demonstrates a path that makes it back to the origin on the final step but is still a failure path since it touched the red line before the final step. To make this image more tenable, we rotate it so that we may assign coordinates to each of the bubbles.

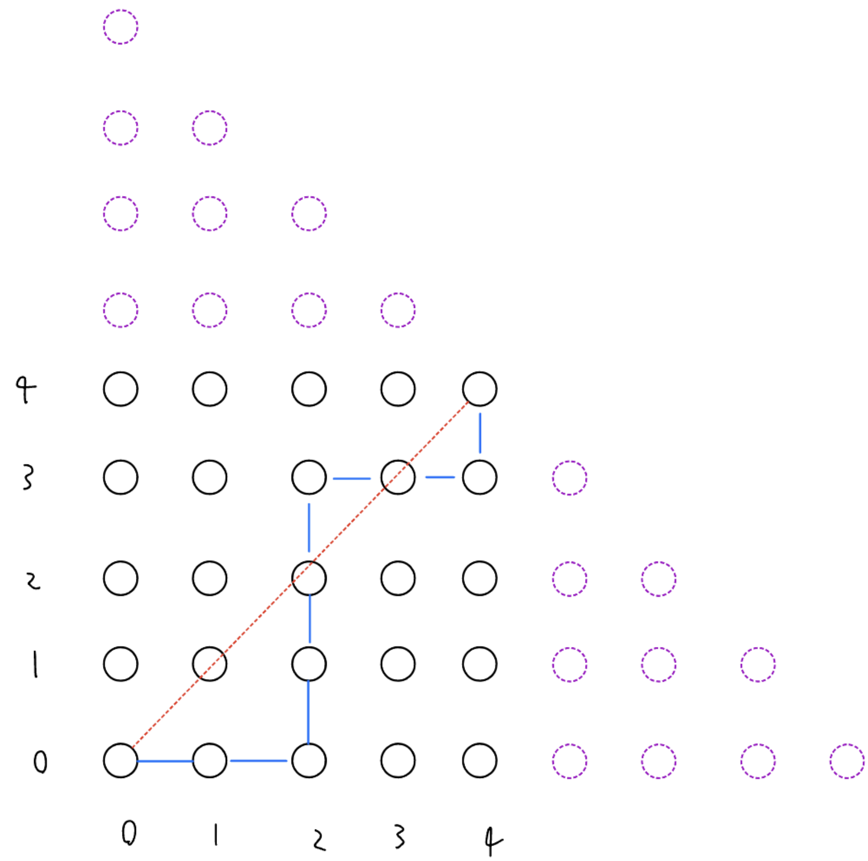

Figure 2: Rotated version of previous image.

Note that nothing has changed qualitatively, but we now have the advantage of being able to assign Cartesian coordinate to each bubble, which we may find to be more organized. Now, a positive step is reflected in an upward step on the graph above, while a negative step is reflected in a rightward step on the graph above. Though, we haven't made any tangible progress on our problem in arranging the paths this way, we now have an image that makes it incredibly easy to observe whether or not a path is a failure. Since we've made the process of failure identification easier, it makes sense for us to pursue this problem through the lens of complementary counting. As such, our focus will be toward counting the number of failure paths.

One simplifying observation we can make is that our setup is symmetric about the line $y = x$ signifying the red line. This means that the number of failure paths originating from the bottom half of the triangle must be equal to the number of failure paths originating from the top half. Therefore, we constrain our focus toward counting the number of bad paths originating from the bottom half below the red line. This implies that all the paths we count will start from the point (1,0).

However, there are still an intractable number of cases to count. To simplify the problem further, we constrain our view to the very last steps that the drunkard can take. The drunkard may only arrive at the origin on his $2n$th step if he was previously at the point $(n-1, n)$ or $(n, n-1)$. Notice that since our shifted origin of (1,0) is below the red line and the point $(n-1, n)$ is above the red line, the drunkard *must* have crossed the red line ($x = y$) on his journey to $(n-1, n)$. Therefore, any paths from our shifted origin that end up at $(n-1, n)$ must be failure paths. We illustrate an example of such a path in the figure below.

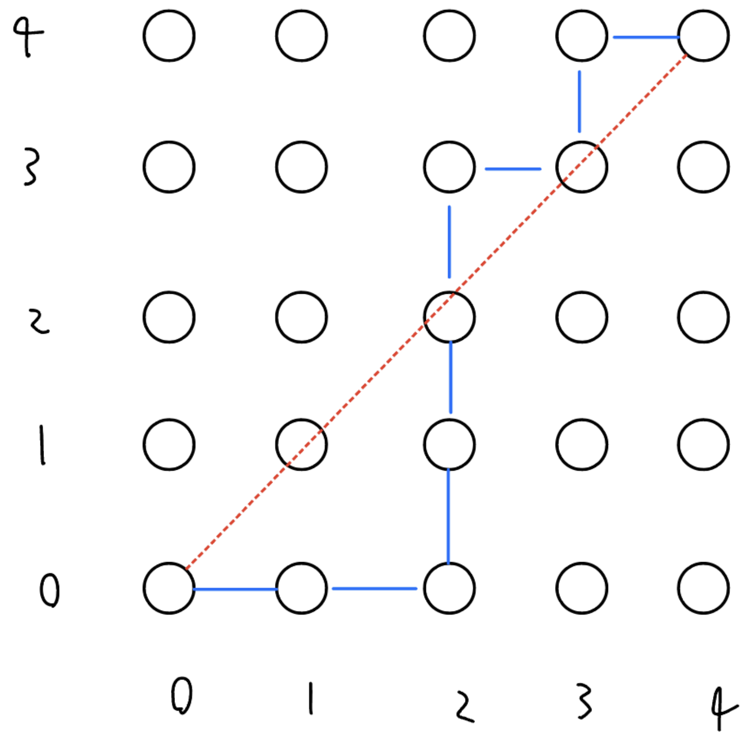

Figure 3: Example of failure path passing through (1,0) and making its penultimate step from $(n-1, n)$ where $n = 4$ in the illustrated case.

To count this quantity, we note that the process of counting paths is equivalent to the process of counting distinct arrangements of n Rs and n Us where an R signifies a step to the right and a U signifies a step upward. An example applicable to our current case would be

$$RRUUURUR$$

Given that we've constrained our paths to start from (1,0) and end up at $(n-1, n)$, the number of Rs to be arranged is $n-2$ and the number of Us is n. Therefore, the number of failure paths from (1,0) to $(n-1, n)$ is

$$f_1 = \binom{n + (n-2)}{n} = \binom{2n-2}{n}$$

Unfortunately, we are still left to count the paths originating from (1,0) and passing through $(n, n-1)$. This counting is even more difficult since there are many ways for the drunkard to cross over the red line twice and end up back under the red line. The path in Figure 2 is an example of this tricky type of walk. Counting these paths is much more difficult since we've lost the ability to say with certainty whether a path has crossed the red line. This is why the previous set of failure paths was so much easier to count. It is here where we can really take advantage of our graphical setup by appreciating its *symmetry*. Let's think outside the box (or outside the triangle more appropriately) and imagine a twin path beginning at (0,1). This path will be a mirror image of our original path reflected about the red line. We provide an example of such twin paths in the figure below.

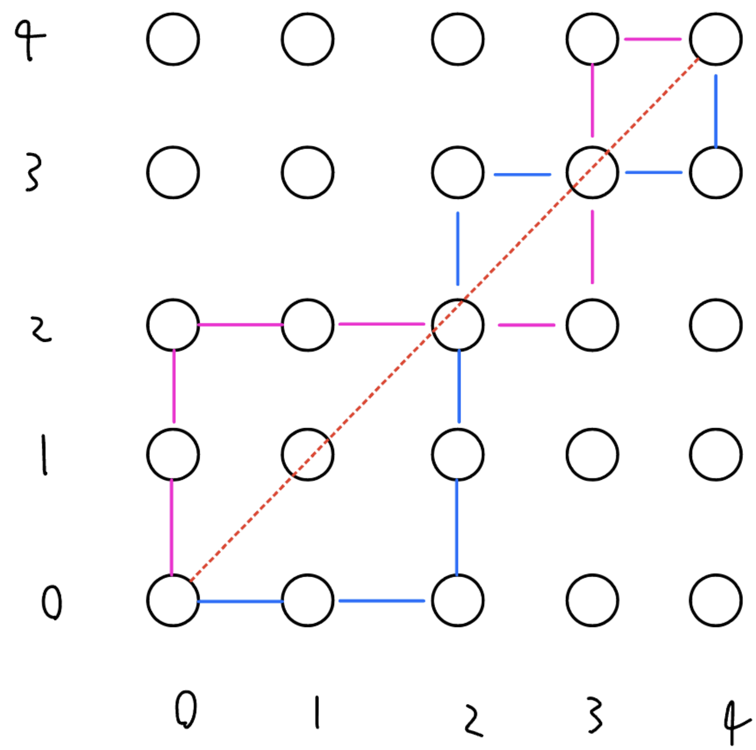

Figure 4: Example of twin paths described above. The blue line demonstrates the second class of failure path, and the pink line acts as its reflection.

By symmetry, we can conclude that if the blue line crosses the symmetry axis, then the pink line must as well. Therefore, there is a one-to-one correspondence between any failure path from the blue line and a failure path from the pink line. To be more concise, whenever the blue line fails, the pink line fails. While this use of symmetry does produce pretty pictures, it may still be unclear how we can exploit this property. The advantage becomes clear once we consider the fact that if both of these lines touch the symmetry axis, then we may simply follow either line to equivalently count the number of failure paths for both cases. We illustrate this property in the figure below.

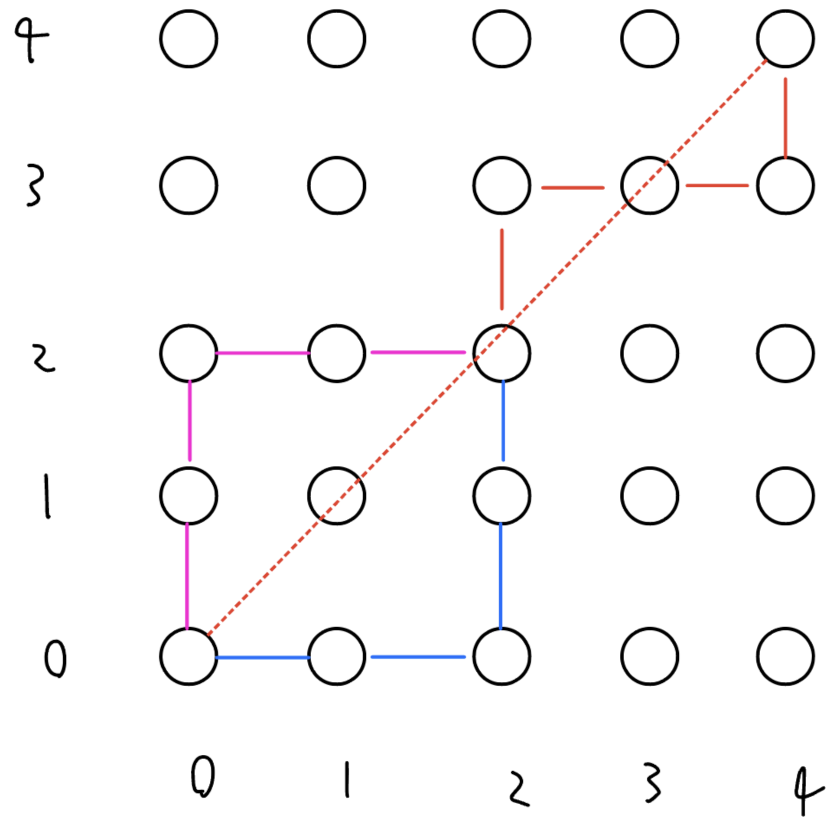

Figure 5: Example of how a failure path may be counted using this symmetry property. This is an example of our desired type of failure path that passes through (1,0) and $(n, n-1)$.

Now we reconsider our case of interest. We are trying to count the (blue) paths that touch the symmetry axis and pass through (1,0) and $(n, n-1)$. However, through the correspondence property we discussed above, we can equivalently count the number of pink paths that touch the symmetry axis and "follow" the blue line to $(n, n-1)$. But, this is simply the number of ways to get from $(0, 1)$ to $(n, n-1)$! This is because $(0, 1)$ and $(n, n-1)$ are on opposite sides of the symmetry axis, so the pink line is forced to cross in order to reach the desired point. Therefore, the number of failure paths of this particular class is again

$$f_2 = \binom{n + (n-2)}{n} = \binom{2n-2}{n}$$

Since every path that returns to the origin must come from $(n, n-1)$ or $(n-1, n)$, the total number of failure paths is $f = 2(f_1 + f_2) = 4f_1$. Note that we introduce the factor of two since we only counted the bottom half through this whole process. By symmetry, there are an equivalent number of failure paths from the top half of the triangle. We recall that the total number of walks that the drunkard can make is 2^{2n}. Thus, by the principle of complementary counting, the probability of the drunkard returning to the origin for the first time on the $2n$th step is

$$p_f = \frac{\binom{2n}{n} - 4f_1}{2^{2n}}$$

$$p_f = \frac{\binom{2n}{n} - 4\binom{2n-2}{n}}{2^{2n}}$$

$$p_f = \frac{1}{n \cdot 2^{2n-1}} \binom{2n-2}{n-1}$$

where in the last step I elected to simplify the result into a single binomial coefficient without displaying the tedious algebra. We verify this result using a simple Python Monte Carlo simulation that I've listed at the end in an appendix. For $n = 20$, I obtained $p_f = 0.0032$ from both the analytic expression and the Monte Carlo.

There is an interesting relation between this result and p_{2n} that we derived earlier. Consider the recurrence $p_{2n-2} - p_{2n}$. If we evaluate this recurrence explicitly, we obtain

$$\frac{\binom{2n-2}{n-1}}{2^{2n-2}} - \frac{\binom{2n}{n}}{2^{2n}}$$

$$\frac{1}{2^{2n}}\left(4\frac{(2n-2)!}{(n-1)!(n-1)!} - \frac{(2n)!}{n!n!}\right)$$

$$\frac{(2n-2)!}{2^{2n}(n!)^2}(4n^2 - 2n(2n-1))$$

$$\frac{2n}{2^{2n}}\frac{(2n-2)!}{(n!n!)}$$

$$\frac{1}{n \cdot 2^{2n-1}}\binom{2n-2}{n-1} = p_f$$

This is quite an interesting result given that we used completely different tactics to approach each problem. Yet, somehow the two results are very cleanly related. To explain why this relation takes the form it does, let's consider how we originally determined p_f.

$$p_f = \frac{\binom{2n}{n} - 4\binom{2n-2}{n}}{2^{2n}}$$

In essence, this result expresses the idea that the probability of returning to the origin for the first time at the very end is p_{2n} minus the probability of the drunkard ending up at either of the points $(n-2, n)$ or $(n, n-2)$ which we will call p_{bad}. If we equate this to the recurrence we previously discussed, we obtain

$$p_{2n} - p_{bad} = p_{2n-2} - p_{2n}$$

$$p_{2n} = \frac{1}{2}(p_{bad} + p_{2n-2})$$

The expression above reveals part of the intuition of the recurrence. It effectively states that half the time, the walker will return to the origin from the corner diagonally to its left, and half the time, the walker will return to the origin from some point two spaces below or to the left of it.

There is one last interesting result that we can produce from our recurrence. Consider the sum below

$$\sum_{i=1}^{\infty} p_f = \sum_{i=1}^{\infty} p_{2i-2} - p_{2i} = (p_0 - p_2) + (p_2 - p_4) + \dots$$

This sum naturally converges to the value p_0. However, p_0 is just 1! This result implies that given an infinite amount of time, the drunkard is *guaranteed* to return to the origin. In fact, this result stretches further. Let's consider a drunkard in the middle of his stroll at some x far away from the origin. Our previous result implies that the drunkard, even from this faraway x, is still guaranteed to return to the origin. We could've redefined the "origin" at his current location x, in which case he is guaranteed to return to $-x$. But we have established no condition for x, which implies that the drunkard is guaranteed to go **everywhere**. This highlights a famous concept referred to as the gambler's ruin. Suppose that a gambler begins with a certain amount of money and plays a fair game betting one dollar each time. His stash of funds will therefore take a random walk. But from our previous result, at some point the gambler will have zero dollars and will be unable to play anymore. Thus, even for a fair game, the house always wins[2].

[2]assuming infinite money for the house

Related Problems (hints in the footnotes)

1.

Consider the setup from above. Show that the probability of not returning to the origin at any time during a finite time interval is also equal to p_{2n}.[1]

2.

Consider a cookie jar with two types of cookies, chocolate chip and raisin. There are x chocolate cookies and y raisin cookies at the beginning with $x \geq y$. You remove the cookies from the jar one at a time and keep track of how many of each type of cookie you've picked. What is the probability that during this process, the number of chocolate cookies you've picked is always greater than the number of raisin cookies?[2]

3.

How many sequences of well-formed parentheses may be constructed from n pairs of parentheses? Well-formed parentheses sequences are those that always have the number of (greater than or equal to the number of) as the sequence is formed. This should be reminiscent of the previous problem.[3]

4.

A pirate has disobeyed his captain by raiding the ship's cellar and drinking all the rum. As punishment, he is sentenced to walk the plank. The pirate is blindfolded and placed on the plank n steps away from the ship. The edge of the plank is N steps away from the ship and stands above shark-infested waters. The captain, being a merciful leader, orders that if the pirate can navigate his way back to the ship from his current location on the plank, then he is absolved of all wrongdoing. However, the pirate is blind and drunk, so he has little sense of direction and will perform a random walk on the plank. What is the probability that he wanders back to the ship and survives?[4]

5.

We build on the previous problem but reintroduce it in the language of betting and coin flips. We will both play a coin flipping game. I start with 60 dollars and you start with 30 dollars. You will flip an *unfair* coin that lands on heads with probability p. If the coin lands on heads, I will give you a dollar. If it lands on tails, you will give me a dollar. The game will end when one of us is broke. What is the *minimum* value of p for which you'd be willing to play this game? In other words, for what value of p is the game fair?[5]

---

[1] Use the recurrence we derived.

[2] This problem may seem completely unrelated to what we previously discussed, but the graphical setup of connected bubbles may be used in the same way. The symmetry argument also applies. On a side note, this problem illustrates a concept called the Ballot Theorem which gets its name from an identical problem.

[3] Once again, the bubble graph setup may be used with the corresponding symmetry arguments. This problem introduces an interesting class of number called the Catalan numbers that have a wide variety of uses in combinatorics and probability.

[4] This problem has quite a different flavor from what we've previously done and may require an approach using the concept of a Markov chain. You may essentially label the probability of the pirate surviving from a given location on the plank as p_i and note that the pirate has equal chance of stepping left or right. Thus, you obtain a solvable recurrence $p_i = 0.5 * p_{i-1} + 0.5 * p_{i+1}$

[5] The process is very similar to the solution of (4.) except the step probability is no longer 1/2 and must be treated generally. Again, solve the general recurrence $a_i = p \cdot a_{i-1} + (1-p) \cdot a_{i+1}$ and note that $a_0 = 0$ and $a_{90} = 1$.

Appendix: Monte Carlo Code

```
import random

N = 1000000 #number of samples
n = 40 #this actually represents 2n as defined in the problem
p = 0 #success counter
coin = [-1, 1] #simulate steps with a coin flip

for i in range(N):
    x = 0
    for j in range(n):
        step = random.choice(coin)
        x += step
        if x == 0 and j < n - 1: #check if drunkard made premature visit
            break
        if x == 0 and j == n - 1: #check if he made it on last step
            p += 1

print(p/N) #final probability estimate
```