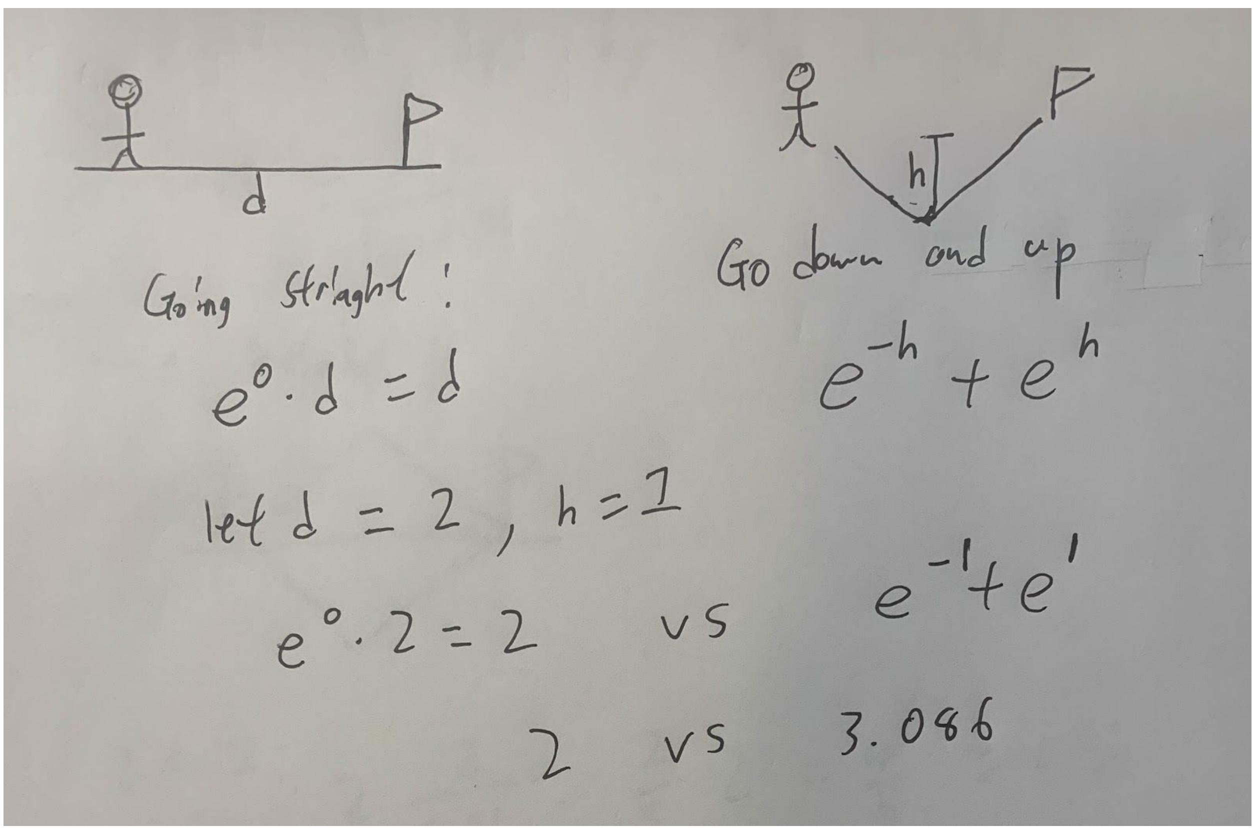
Approach

For creating an admissible heuristic, I used an approach of breaking down the possible situations with respect to the goal state into different use cases. Then I thought of the best possible situation for each of the cases and created a way of underestimating them.

Exponential function

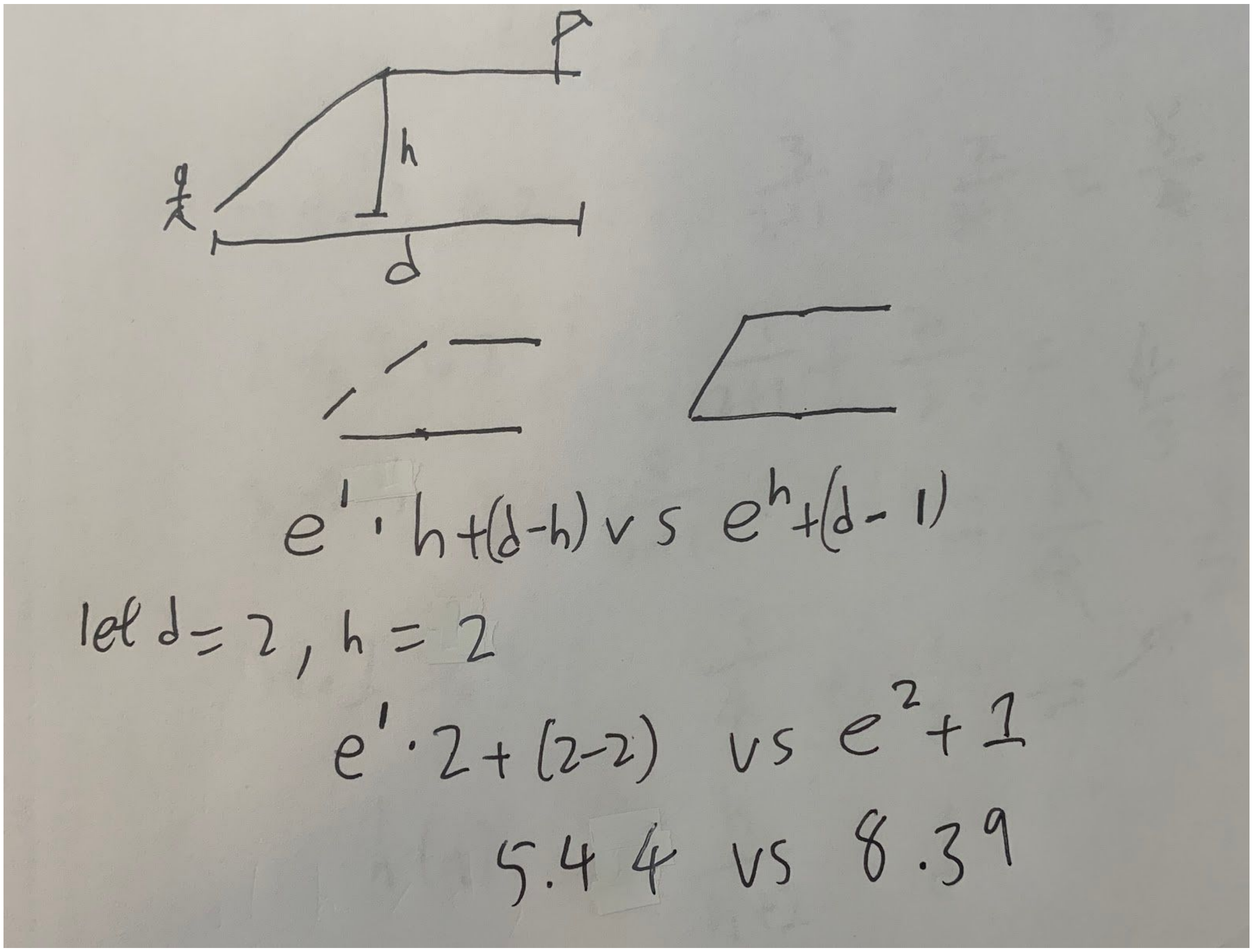
For this function I have broken down the possible situations with respect to the goal state. Let GH be the goal's height, CurH be current height, d be the distance between current state and goal state, and h be GH - CurH. The different situations I have created are, GH == CurH, CurH < GH and h <= d, CurH < GH and h > d, and CurH > GH.

GH == CurH



Here we can see that going straight will always be better than going down and back up again if the goal state's height is the same as the current state's height. I proved that this is true for h = 1. This also holds true when h is greater than 1 since e is an exponential function and it will increase exponentially when h increases. Therefore the h(n) = e^0 *d, which is equal to h(n) = d.

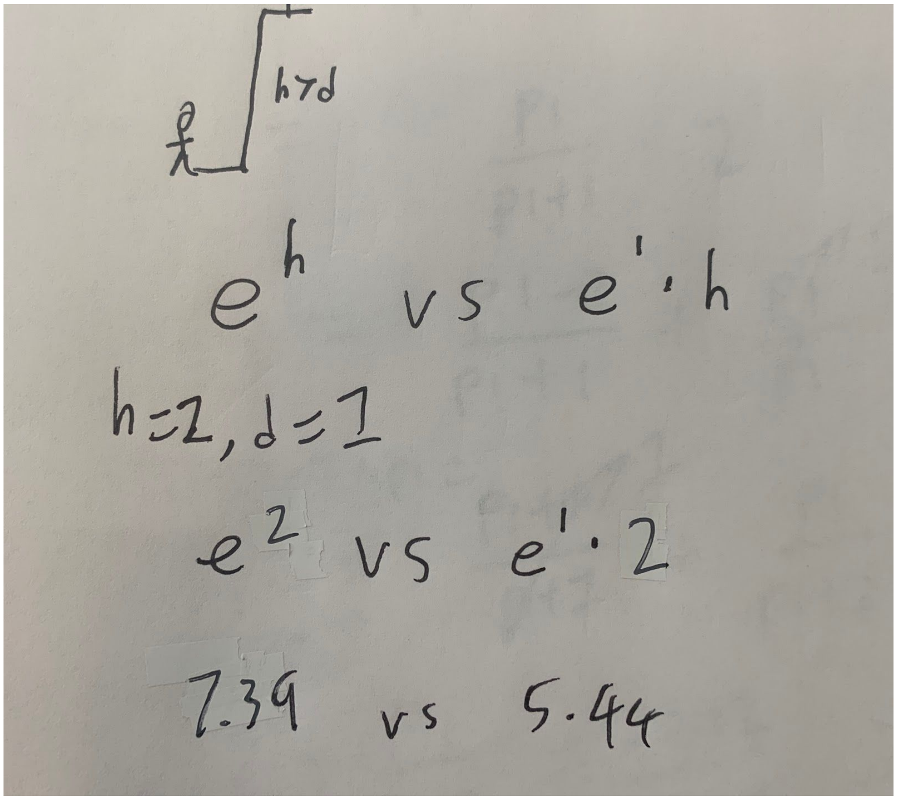
CurH < GH and h<= d



When the goal's height is above the current height and d is greater than or equal to h, we want to compare if taking one step at a time or taking one big step and walking straight after would be preferable.

For this proof I chose a h with 2 since when h = 1, their cost would be the same. They would both be e^1. When d = 2 and h = 2, we can see that indeed taking one step at a time would be the best outcome. This holds true when h increases since e^h -1 will be higher than e^1 *h - h. If d is greater than h, we would want to walk in a straight line from the proof prior for the situation of GH == CurH. Therefore, the h(n) = e^1 * h + (d - h).

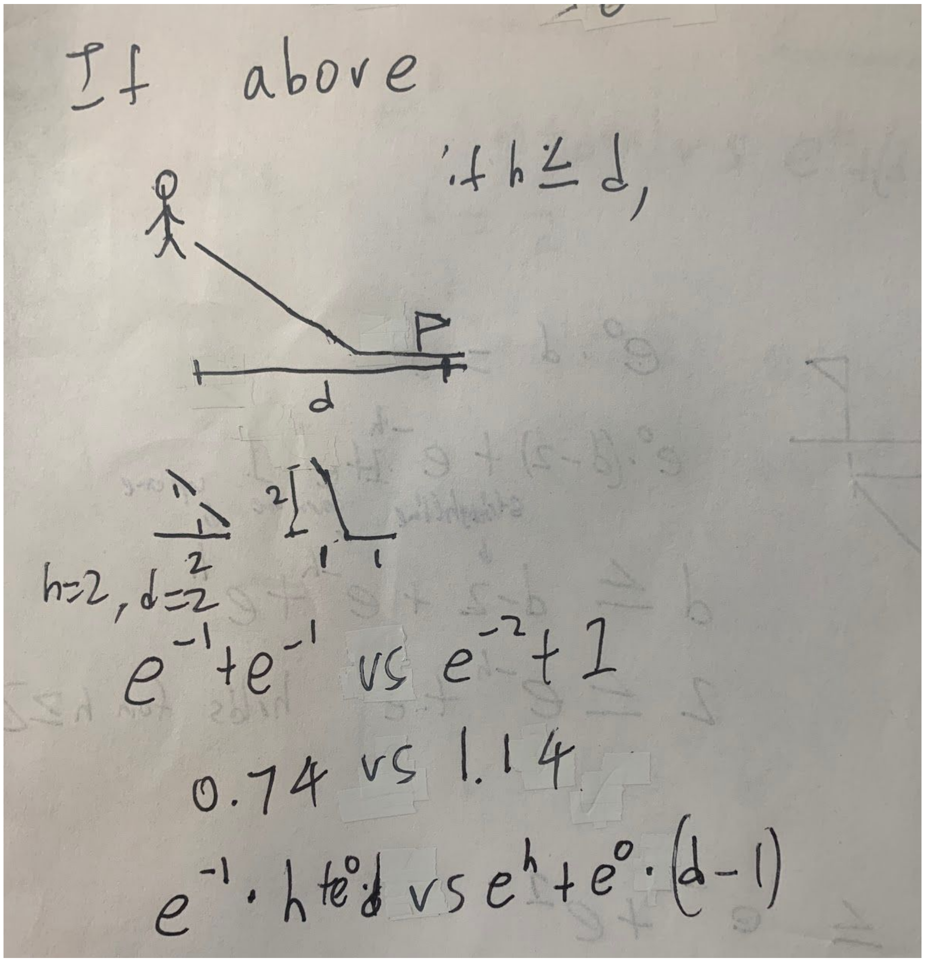
CurH < GH and h > d



When the goal is above the current, but the d is less than h, we would want to compare if taking that big step or considering a spiral staircase would be the preferable outcome.

Here, we can clearly see that taking the path of e^1 * h would be less than e ^h since e is an exponential function and would always be greater than when h is greater than d. Therefore, the h(n) = e^1 *h

CurH > GH



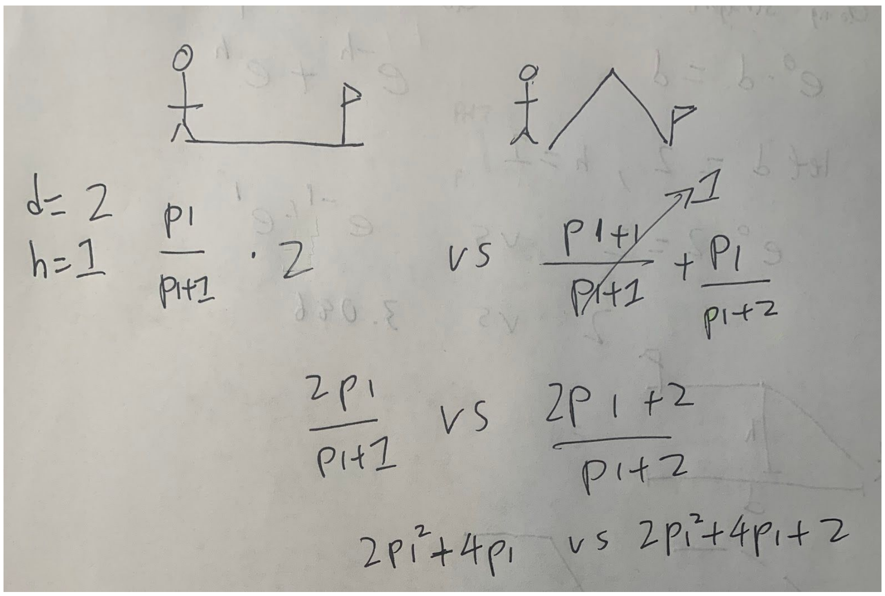
Here we want to compare if taking one step at a time going down would cost less or taking a big step down and walking straight after would cost less.

In the proof, I used h =2 and d=2. It turns out that taking one step at a time would cost less than taking a big step down. Even though e^h going down is less than e^-1, walking straight after would cost d-1 and this outweighs the cost function. Going down to the same height costs e^-1 * h in the best case and then we have to add the difference between d and h for the straight path after the same level is reached. The h(n) = e^-1 *h + (d-h)

New Height divided by old height function

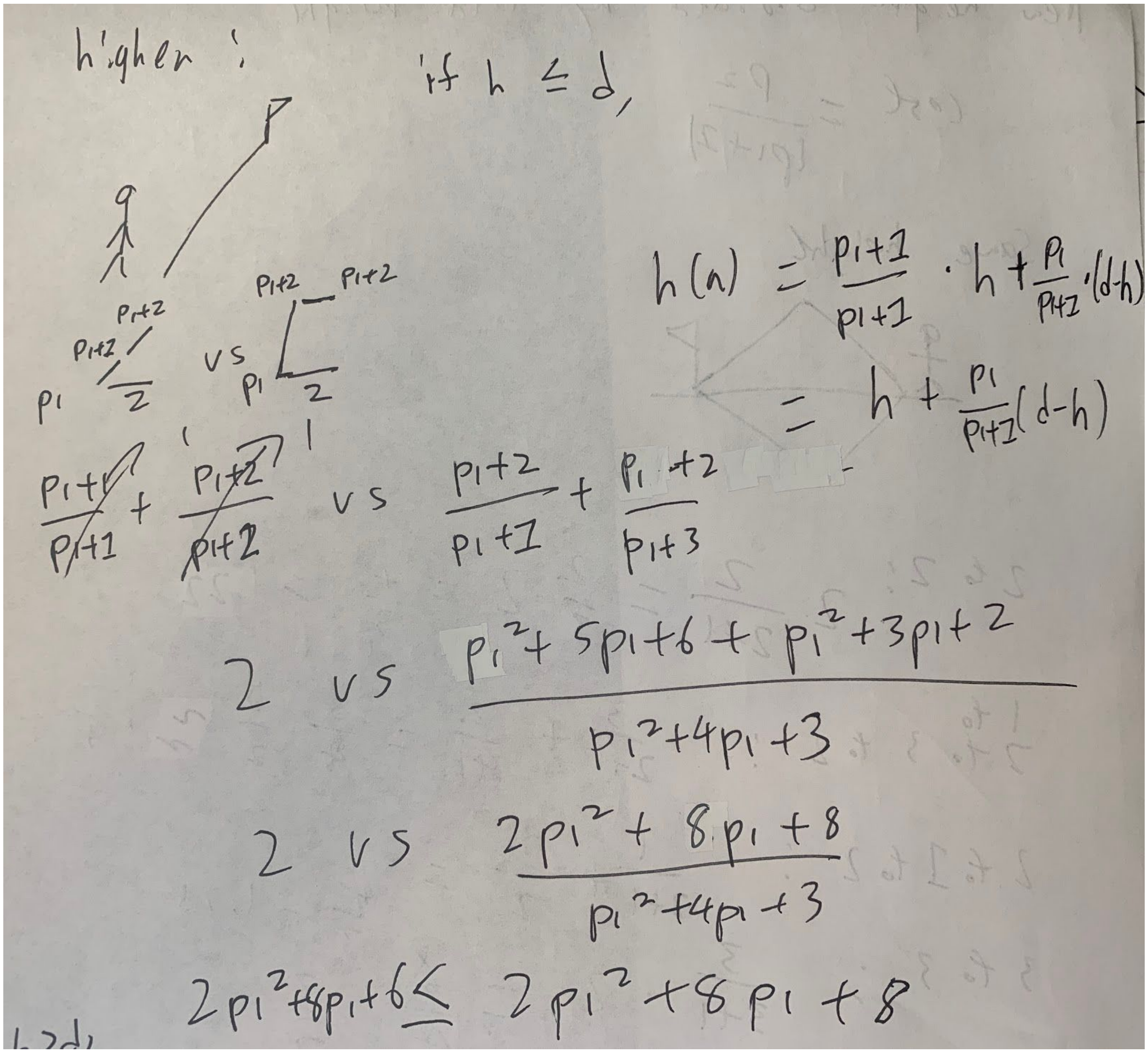
For this function I have broken down the possible situations with respect to the goal state. Let GH be the goal's height, CurH and p1 be current height, d be the distance between current state and goal state, and h be GH - CurH. The different situations I have created are, GH == CurH, CurH < GH and h <= d, CurH < GH and h > d, CurH > GH and h <= d, and CurH > GH and h > d.

GH == CurH



Here, we want to compare if walking straight would be better than going up and down to reach the goal state.
For d = 2 and h =1, we can see that walking straight would cost less than going up and down by 2. Thus, the h(n) = p1 / p1+1 * d.

CurH < GH and h <= d



$$\cancel{\frac{p_1+1}{p_1+1}} + \cancel{\frac{p_1+2}{p_1+2}} \quad vs \quad \frac{p_1+2}{p_1+1} + \frac{p_1+2}{p_1+3}$$

$$2 \quad vs \quad \frac{p_1^2+5p_1+6+p_1^2+3p_1+2}{p_1^2+4p_1+3}$$

$$2 \quad vs \quad \frac{2p_1^2+8p_1+8}{p_1^2+4p_1+3}$$

$$2p_1^2+8p_1+6 \le 2p_1^2+8p_1+8$$

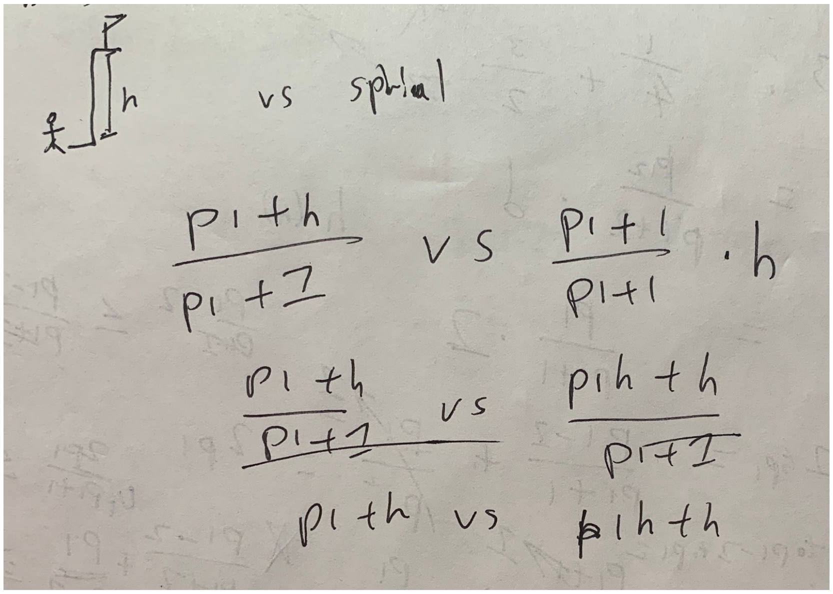
$$h(n) = \frac{p_1+1}{p_1+1} \cdot h + \frac{p_1}{p_1+2} \cdot (d-h) = h + \frac{p_1}{p_1+2}(d-h)$$

Here, we want to compare if taking a step a time or going up a big step and walking straight after would be preferable.

To illustrate this, I have chosen h to be 2 and d = 2. As we see above, taking a step up at a time would cost less than taking a big step and walking straight by a difference of 2. After we have reached the same height as the goal's height, we would want to move straight, proven by the GH == CurH case.

The distance we need to walk straight is d-h because we took one step for each h. Therefore putting everything together, we get h(n) = h + p2 / p2 +1 (d-h).

CurH < GH and h > d



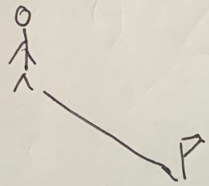
Here, we are comparing taking a big step up or taking a spiral staircase (one step up at a time around the goal) would be preferable.

From the proof, we can see that taking a big step up which costs p1 + h is less than a spiral staircase which costs p1h+h.

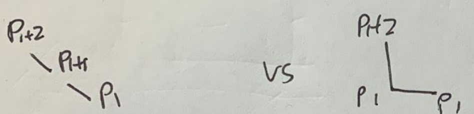
Therefore, the h(n) = p2 / p1+1 + p2/p2+1(d-1).

CurH > GH and h <= d

lower: if $h \le d$



$$h(n) = \frac{p_1 - h}{p_1 + 1} + \frac{p_1}{p_1+1} \cdot (d-1)$$



$$\frac{P_1+1}{P_1+3} + \frac{P_1}{P_1+2} \quad vs \quad \frac{P_1}{P_1+3} + \frac{P_1}{P_1+1}$$

$$\frac{P_1^2+3P_1+2+P_1^2+3P_1}{P_1^2+5P_1+6} \quad vs \quad \frac{P_1^2+P_1+P_1^2+3P_1}{P_1^2+4P_1+3}$$

$$\frac{2p_1^2+6p_1+2}{p_1^2+5p_1+6} \quad vs \quad \frac{2p_1^2+4p_1}{p_1^2+4p_1+3}$$

$$2p_1^4+14p_1^3+32p_1^2+26p_1+6 \quad vs \quad 2p_1^4+14p_1^3+32p_1^2+24p_1$$

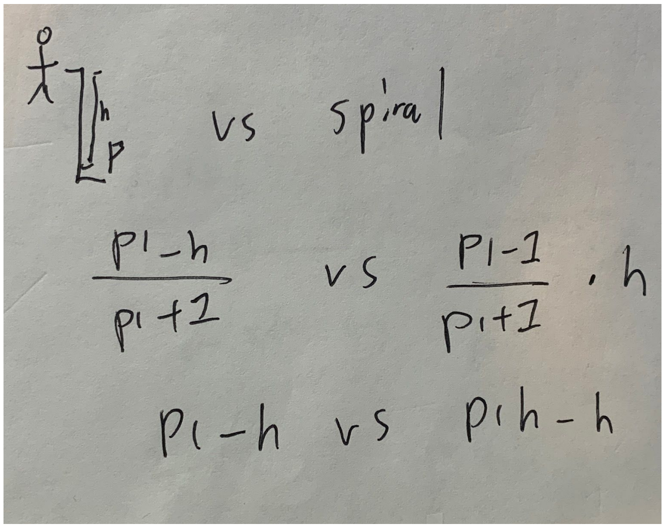
$$26p_1+6 \quad vs \quad 24p_1$$

When CurH > GH and h <= d, we want to compare if taking one step down at a time or taking a big step down and walk straight would be better.

From above, we can see that taking a big step down and walk would be preferable. Big step down and walk costs 24p1, whereas taking one step at a time would cost 26p1 +6.

Once we reached the same height as the goal's height, we would want to walk straight based on the proof from Gh == CurH. Therefore the h(n) = p2/p1+1 + p2/p2+1 *(d-1).

CurH > GH and h > d



Here, we are comparing taking a big step down and taking the spiral staircase down.

We can see from above that taking the big step down costs p1-h which is less than the cost of p1h-h for taking the spiral staircase down.
Therefore the $h(n) = p2/p1+1 + p2/p2+1 *(d-1)$.

Dumb down heuristic for MtStHelensDiv

For the Mount Saint Helens file division function, I modified the heuristic. The reason I did that is because it will require less time to compute the answer thus resulting in better performance.

I got this heuristic by relaxing the rule that a tile can move from A to B if A is the neighbour of B. Then the only thing needed to care about is the distance from the current state to the goal state. Since the map uses chess movement, the cost of d would be found by the chebyshev distance.

$h(n) = Max(|B.x-A.x|,|B.y-A.y|)$

Div:
Seed 1
PathCost, 198.59165501141644, Uncovered, 1004, TimeTaken, 25

Seed 2
PathCost, 198.56141550095256, Uncovered, 1004, TimeTaken, 19

Seed 3
PathCost, 198.4864317411443, Uncovered, 1004, TimeTaken, 22

Seed 4
PathCost, 198.70066424826052, Uncovered, 1004, TimeTaken, 22

Seed 5
PathCost, 198.25499446810397, Uncovered, 1004, TimeTaken, 22

MtStHelens
PathCost, 548.3684300960452, Uncovered, 106827, TimeTaken, 1079

Exp:
Seed 1
PathCost, 533.4482191461119, Uncovered, 71645, TimeTaken, 2985

Seed 2
PathCost, 549.5036346739352, Uncovered, 81880, TimeTaken, 3445

Seed 3
PathCost, 510.97825243663607, Uncovered, 74001, TimeTaken, 2803

Seed 4
PathCost, 560.6570436319696, Uncovered, 66382, TimeTaken, 2645

Seed 5
PathCost, 479.5879215923168, Uncovered, 67837, TimeTaken, 3020

MtStHelens
PathCost, 515.6645805015318, Uncovered, 119087, TimeTaken, 1957