**Required Explorations:**

4) Tuple Random Tie-Breaker

Puzzle: 35 FIEBDA0CONKGHLMJ

path length: 35. seconds to run: 5.9375.dlluruldddrurdlurrdlluuurrdldrullul

Puzzle: 36 GCBFAEHKDI0JLNMO

path length: 36. seconds to run: 32.484375.uulldrruldlddrurdllurrdlurruuldrulll

Puzzle: 37 FIBEALDKJCNGHM0O

path length: 37. seconds to run: 5.171875.uluurdddluldrruruldluurrdlluldrurdlul

Puzzle: 38 IBKFGACNDOMJLHE0

path length: 38. seconds to run: 1.296875.luldrurululdlurrrdllldrrrdluruldlurull

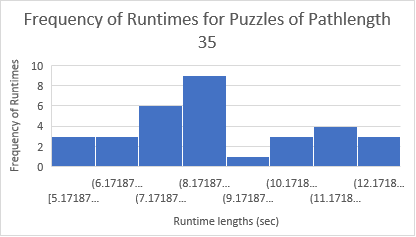
Puzzle: 39 BKNCAIEJD0FMLHGO

path length: 39. seconds to run: 16.828125.urdluurdlddrurullulddrdrulurdldrurulull

Puzzle: 40 MFKCAEOGBH0IDLNJ

path length: 40. seconds to run: 28.203125.uurddlullddrrrulurullldrulddrurddluurull

The A\* algorithm uses a heuristic to calculate the distance of a puzzle to a goal state and because of the sheer number of nodes processed, there is bound to have a “tie” when the distance between the puzzle to a goal state is equal to another puzzle’s distance. So how will this affect A\* when selecting which node to run form the heap? If run repeatedly, the program will always choose the same path. However, when we add a randomized tie-breaker, the program will iterate through random paths. Established that there are multiple solutions of the same pathlength, the random path selection can be illustrated by dramatic differences in runtimes – if the same path was found every time, the runtimes would have little variation. The graph below displays just this. I ran Puzzle 35 from 15\_Puzzles.txt 30 times and the runtimes do vary dramatically, the pathlength accurate every time.



5) Heuristic Multiplier

a) As I increase the value of m, the speed of the runs become slower, but supposedly more accurate. On the other hand, as the value of m decreases, the speed increases, however sacrificing accuracy. It is easy to see the effect of the multiplier on the runtime as displayed below. I ran the A\* program on puzzle 30 using m values from 0.5 to 1.5 incrementing by 0.1 each time. Illustrated below, the runtime increases at a seemingly exponential rate as the multiplier increases. This is possibly true because greater multipliers mean that the node would be ranked lower on the heap than it should (distance from path is reported to be greater than what it actually is). This means it would take longer, by running through more paths, therefore raising the probability of finding the correct one. The opposite is true for smaller m values. As the value decreases, the estimation of distance becomes shorter, therefore running through that path becomes a higher priority – even if it’s is incorrect.

In order to show the advantages of using a random tiebreaker at with a multiplier, I ran Puzzle 50 at multiple times at multiple levels of m. Below is the results of running Puzzle 50 with a m value of 0.7 and random tiebreaking. The results show that random tiebreaking also has the advantage of introducing variability in pathlengths when working in tandem with a multiplier. This allows for a higher chance of returning an accurate estimate of the shortest pathlength with the increased speed of using a small m value. However, a balance between m values and puzzle pathlengths need to be found: too small a m value, the variability is overcome by the prioritization of small estimates of pathlength, too large a m value, the accuracy of A\* would overcome variability as well as increase in runtime. The process of obtaining the chart below proved this, as running on m values less than 0.7 gave the same incorrect solutions, and above 0.8 gave consistent correct solutions with greater and greater runtimes.

6) korf100

Puzzle: 1 NMOGKLIEF0BADHJC

path length: 59. seconds to run: 67.34375.uuldddruuuldddruururddluurdddluluurrdddluululdrurrdldrulllu

Puzzle: 2 MEDJILHNBCGA0OKF

path length: 57. seconds to run: 4.8125.uurrdruldlluurrddrdllluurddrulluurrddruuldddluurrddluullu

Puzzle: 3 NGHBMKJDILE0CFAO

path length: 61. seconds to run: 32.75.lulddruurdlldluurruldddluuurdrrdlldluurrdrulurdldlldrurdlluuu

Puzzle: 4 ELJGOKN0HBAMCDIF

path length: 60. seconds to run: 11.015625.lldrrullddluuruldrddluurddlurrdlurrdluruuldlurddruuldrdluull

Puzzle: 5 DGNMJCILKEFOABH0

path length: 60. seconds to run: 8.0625.uluuldddruurullddldrrulluurddlurrrdllurrullldddruuuldddruluu

Puzzle: 6 NGAILCFOHKBEJ0DM

path length: 52. seconds to run: 7.875.rruullurrdldlluurrdlddrruldlluruldruurrdldldrulurull

Puzzle: 7 BKOEMDFGLHJAICN0

path length: 54. seconds to run: 30.765625.lllurdluurrurdllurdldldrurrulddruuldluurdrullldrruldlu

Puzzle: 8 LKOCH0DBFMIENAJG

path length: 52. seconds to run: 41.46875.ldruruldlurdddluurdruuldddruurddlluurrdldlluuurdrull

Puzzle: 9 CNIKEDHBMLFGJAO0

path length: 48. seconds to run: 2.4375.llluurulddrurrdldluuruldrddllururrullddrruldlulu

Puzzle: 10 MKHI0OGJDCFNELBA

path length: 61. seconds to run: 28.5.urrrdldlulddruuuldrrddluuurddrdluruuldldrurdlululdrdluurddluu

Puzzle: 11 EIMNFCGLJHD0OBKA

path length: 59. seconds to run: 0.4375.ullurddrdlllurrruuldddluuurddlulddruluurrdrddllurulddrululu

Puzzle: 12 NAIFDHLEGBC0JKMO

path length: 45. seconds to run: 0.65625.lllurrdluldrdluuurrdrdlluldruurrddlldrulurull

Puzzle: 13 CFEBJ0ONADMLIHKG

path length: 50. seconds to run: 27.53125.ldrrrdluululdddrrulurrddluuldldrruulurrdllurdrulll

Puzzle: 14 GFHAKENJCDIMOB0L

path length: 65. seconds to run: 438.578125.ruululddldrrullurrdruldlldrrrulluulddrdrruluuldrurdllurdllurdrull

Puzzle: 15 MKDLAHIOFENBGCJ0

path length: 64. seconds to run: 13.265625.luruldluurdlldrdluuurdddrruulldrrulldluurrrdlulddruldrdluldruuul

Puzzle: 16 ACBEJIOFHNMKLDG0

path length: 46. seconds to run: 18.90625.luuuldddruruuldlldrrurddlulurrdlluldruulddrulu

Puzzle: 17 ON0DKAFMGEHICBJL

path length: 72. seconds to run: 131.375.lldddrurruulllddrrurullldrrddruuullldrddlurdrruuullldddrrrullurdruuldllu

Puzzle: 18 F0NLAOIJKDGBHCEM

path length: 59. seconds to run: 12.0.lddrururddlurdlldrrulluldrurdldrruulurddluulddrdluldrruuull

Puzzle: 19 GKHCN0FOADMIELBJ

path length: 46. seconds to run: 0.34375.ldruurdluldddrurdruulddruldluuurdlulddrrrulllu

Puzzle: 20 FLKCMGIOBNHJDAE0

path length: 54. seconds to run: 8.703125.ulldruulldruuldddrurrululldddruruullddruldrdrruldlluuu

Puzzle: 21 LHNFKDG0EAJOCMIB

path length: 56. seconds to run: 51.0625.lldldrrulluurddluurdrdrdluldluurrurddluuldrrddluruuldlul

Puzzle: 22 NCIAOHDEKGJM0BLF

path length: 61. seconds to run: 45.875.rrulldruuldrurdldrurululldrurrdddlulurrdlldrulurulldrrrulldlu

Puzzle: 23 JICK0MBNEFDGHOAL

path length: 51. seconds to run: 23.1875.druuldrrddlurruulldrdrdlurululddluurddrdluldrruullu

Puzzle: 24 GCNMDAJHELIKBOF0

path length: 54. seconds to run: 10.203125.lluruulldddrulurdrurulldrrddlululdrdruuulddrrulddluulu

Puzzle: 25 KDBGA0JOFINHCMEL

path length: 52. seconds to run: 5.171875.lurdlddrrrululldrdrruullldrdrruullldrdlurulurrdrulll

Puzzle: 26 EGCLOMNH0JIFADBK

path length: 60. seconds to run: 34.609375.rulddruulddrurdlururddluuldruurdllurdlulddruurrdlullddrdluuu

Puzzle: 27 NAHOBF0CILJMDGEK

path length: 55. seconds to run: 60.390625.ddluuuldrurddluuldddruurrulddruulldldrrdruldluurrdlulul

Puzzle: 28 MNFLDEA0ICJBOKHG

path length: 54. seconds to run: 14.0625.uldrdllurddlluruulddrrrdllluruurrdlurdddlluurddlurullu

Puzzle: 29 IH0BOADNCJGEKMFL

path length: 56. seconds to run: 19.53125.ddlluurrddruldlluurdrddruulurdldldlurdrrullldrrruldluluu

Puzzle: 30 LOBFANDHECG0JMIK

path length: 51. seconds to run: 1.609375.lluuldddruurrdlldrullurrrddluluurrdlullddruldrdluuu

Puzzle: 31 LHOMA0EDFCBKIGNJ

path length: 52. seconds to run: 0.078125.lurddluurdrrulldrdldluurdrruulddrullddrruulldrdluuul

Puzzle: 32 NJIDMFEHBLG0ACKO

path length: 61. seconds to run: 11.484375.ldluuldruuldddruuurdrdldluurdluldrdrurullurrddlluurdluldrrull

Puzzle: 33 NCEOKFMI0JBLDAGH

path length: 60. seconds to run: 83.96875.rrdruullddrruuullddldrrruuulldlurrdllddrrrulurdllurdldruuull

Puzzle: 34 FKGHMBEDAJCIN0LO

path length: 52. seconds to run: 3.296875.ulurrrullldrrddlluurdrruuldrulllddrrrululddrdllurulu

Puzzle: 35 AFLNCBOHDEMI0GKJ

path length: 55. seconds to run: 35.71875.rruuuldddruurullddrulldrrruldrdlllurdruuldrrullurrdlull

Puzzle: 36 LF0DGCOAMIHKBNEJ

path length: 54. seconds to run: 12.828125.rdlldrrulddllurulurrrdllddrruuldlluurrdddllurulurrdlul

Puzzle: 37 HAGLK0JEIOFMNBCD

path length: 58. seconds to run: 102.515625.rddruuulddlulurrdrdlulddruulddrrulurullddldrulurrdldruuull

Puzzle: 38 GOHBMFCLK0DJIEAN

path length: 53. seconds to run: 1.5625.rdruulldlurrdluurdluldrurrdldrullddrruulldrdluldruluu

Puzzle: 39 I0DJANOCLFEGKMHB

path length: 49. seconds to run: 1.625.rrdlllurddrdllurrurdlulldrdrurdlulururddluruldllu

Puzzle: 40 KEANDLJ0BGMCIOFH

path length: 56. seconds to run: 8.578125.dlldrrulluulddruurrddlullurdddruuuldddlurdrrululddruluul

Puzzle: 41 HMJIKCOF0ABNLEDG

path length: 54. seconds to run: 6.859375.uurddruulddldrrurdluldluuurdrruldrdlulddluurdldruruull

Puzzle: 42 DEGBINLM0CFKHAOJ

path length: 42. seconds to run: 0.9375.rrurddluuldlurddruuldldrrurululdlurrrdllul

Puzzle: 43 KONMAIJDCFBLGEH0

path length: 70. seconds to run: 4.765625.ulurdllurulldrrrulddruullldddrrruulllddrrrulullddrrurulurdlldrruldlulu

Puzzle: 44 LI0FHCENBDKGJAOM

path length: 50. seconds to run: 14.71875.dldluurddluurrdrulldddluurdruuldrrdldrulldrrullluu

Puzzle: 45 CNIGLO0DAHEFKJBM

path length: 53. seconds to run: 1.34375.lldruulddrrurdllurddrulldluurddruluruldrrdllurruldlul

Puzzle: 46 HDFANLBOMJIECG0K

path length: 51. seconds to run: 8.921875.llurrrulddlluurrddruluurddldlluuurrrddldllurrdluluu

Puzzle: 47 FJANOHCEM0BGDIKL

path length: 47. seconds to run: 1.03125.uurdrulddlulurrdldrurdldrululddrulldruuldrrulul

Puzzle: 48 HKDFGCJIBLOM0AEN

path length: 53. seconds to run: 3.015625.ruldrrurdluulurrdlllurrrdlldrrululdddruuldlurdruuldlu

Puzzle: 49 J0BDEAFLKMIGOCNH

path length: 59. seconds to run: 31.96875.rdrulldlurdrrddllluurrdrdllulurdrrdlllurdrulurrdlluurrdlull

Puzzle: 50 LEMKBJ0IGHDCNFOA

path length: 57. seconds to run: 9.171875.llurddlurrdlurrddllluurrurddluulddrurdldluurdllurulddruul

Puzzle: 51 JBHDO0ANKMCFIGEL

path length: 58. seconds to run: 2.3125.ruldrddluuldrrrulllurrrdlldrdruuldldrruululdlddruurdllurul

Puzzle: 52 JH0LCGFBANDKOMIE

path length: 58. seconds to run: 17.765625.rdlllddrurululdrdlururrdldldruluurrdddluruldlurullddrdluuu

Puzzle: 53 NILMODHJ0BAGCKEF

path length: 70. seconds to run: 61.984375.urrdluuldrurdddluldruuurddlurruldrddlluruurdddlullurdldrruullurrdrulll

Puzzle: 54 LK0HJBMOEDGCFINA

path length: 58. seconds to run: 17.328125.ldrrddluuurddldrulurdlluruldlddrurullurddlurdrdllurdruulul

Puzzle: 55 MHNCIA0GOEDJLBFK

path length: 43. seconds to run: 1.25.dlluurdruldrddluulurrdddluurdrdlluluurrdlul

Puzzle: 56 COBEKFDGLIA0MNJH

path length: 55. seconds to run: 149.0.dllluurrulldrdrrululldrdruurdlulddlurrurddlldrruldluluu

Puzzle: 57 EKFIDML0HBOJAGCN

path length: 52. seconds to run: 3.859375.dllurddluuurdddluldrrruullurrdldrullldrruulldrrdluul

Puzzle: 58 E0OHDFANJKCIGLBM

path length: 51. seconds to run: 9.546875.drdruuldrullddldrruulddrruulldluurddrulldrdrruulull

Puzzle: 59 ONFGJA0KLHDIBEMC

path length: 57. seconds to run: 255.390625.druulldlurdlddruurdllurddlurdruuulddrurddluurdluurddluull

Puzzle: 60 KNMABCLDOGIEJFH0

path length: 70. seconds to run: 91.53125.ululdrdllurrrullurrdlddlluruurrdddlluurrddlluuuldrrrullldrrurdldruulll

Puzzle: 61 FMCBKIEJAGLNHD0O

path length: 45. seconds to run: 11.46875.ulurdrullldrdruruuldlulddrulddrururdlluurdlul

Puzzle: 62 DFL0NBIMKHCOGJAE

path length: 61. seconds to run: 14.390625.llddlurrddruuulddlurrdlurullddrdllururddllururddrulurddlluluu

Puzzle: 63 HJIKNAGOMD0LFBEC

path length: 56. seconds to run: 465.84375.ruullddrdruulddlluuurrdddllurdluururddruuldrdldluurdluul

Puzzle: 64 EBN0GHFCKLMODJIA

path length: 53. seconds to run: 70.203125.ddldluldrrruuldlurddruululdldrrulddruruldllurulddruul

Puzzle: 65 GHCBJLDFKMEO0AIN

path length: 47. seconds to run: 3.453125.ruldrrrulluldruuldrrruldrdlldluurrrulldrurdlull

Puzzle: 66 KFNLCEAOH0JMIGDB

path length: 63. seconds to run: 138.421875.rdluurdruullldrrrullldrdrrdluurddluuuldddruuruldrdlluuldddruluu

Puzzle: 67 GABDHCFKJO0ENLMI

path length: 50. seconds to run: 44.59375.lldrrurulldluurrrdldlulurrrdddluululdrdrrulldruull

Puzzle: 68 GCAMLJEBH0FKNODI

path length: 53. seconds to run: 36.625.rdluurdllurruldrruldrddllluuurdldrruldrrulddluurdlulu

Puzzle: 69 F0EOANDIBMHJKLGC

path length: 57. seconds to run: 31.625.lddrdlururrullldrrrddllururddluuurdddllluruurdluldrruldlu

Puzzle: 70 OACLD0FEBHNIMJGK

path length: 54. seconds to run: 27.84375.rrulllddruurdrdluldrdrulurdldrululldrrulddluuurrddluul

Puzzle: 71 EG0KLAIJOFBCHDMN

path length: 46. seconds to run: 10.09375.ldrdllurrdldluuurdrrdluruldlldrdrruuulldldruul

Puzzle: 72 LOKJDEN0MGABIHCF

path length: 58. seconds to run: 94.125.ldruuldlurdldrdruuldldluuurrddruuldlldrrdruuldluulddrdluuu

Puzzle: 73 FNJEOHGACDB0LIKM

path length: 53. seconds to run: 0.4375.lluldruurrdldlurulldrrurddldrulldruuulldrruldrruldllu

Puzzle: 74 NMDKOHFI0GCABJLE

path length: 66. seconds to run: 20.703125.ururddlluurrddruldlurrulldldrdrurdluruldldlururdldruulddluuurdrull

Puzzle: 75 ND0JFEACIBMOLGHK

path length: 48. seconds to run: 21.609375.dldrdlurdruldlluuurdldrururddlluruldlddrruuldluu

Puzzle: 76 OJHC0FIEANMKGBLD

path length: 61. seconds to run: 26.9375.ddrruululdrurdldruldldrruurddlululdrurdruldlulurdruldlurddluu

Puzzle: 77 0MBDLNFIOAJCKEHG

path length: 58. seconds to run: 12.328125.rdddluurrddllurrululddruurdrddlluururdddlluulurrdluldrrull

Puzzle: 78 CNMFDOHIELJ0BGAK

path length: 57. seconds to run: 0.484375.ldluurdluulddruruldlddrulurrrddlulurdruuldrddlluurrdlullu

Puzzle: 79 0AIGKMECNLDBHFJO

path length: 42. seconds to run: 1.6875.rrdddluldruuldrurdllurrrdldluuurdrulddluul

Puzzle: 80 K0OHMLCEJADFNIGB

path length: 61. seconds to run: 2.875.ldrdrrululddruurddldruluulddluurdrrdldlluurrdllururdddlurullu

Puzzle: 81 M0ILKFCEOHAJDNBG

path length: 55. seconds to run: 0.734375.rrdlddluluurrdlurrdldlulddruulurrdllddrrurdllururdlluul

Puzzle: 82 NJBAMIHKGCFLOED0

path length: 64. seconds to run: 711.328125.lullurrdrululdrurdllddluuurrdldrdruullldrurddlurrdllluuurddlurul

Puzzle: 83 LCIADEJBFKO0NGMH

path length: 51. seconds to run: 25.171875.lluldrdrruluuldddruuurdlllurrrdllldrdluruldrrdlluuu

Puzzle: 84 OHJG0LNAEIFCMKDB

path length: 57. seconds to run: 10.21875.urrrddldluuldrruurdddluruldluurddldrurdlllurulurdrdllurul

Puzzle: 85 DGMJABIFLHNEC0KO

path length: 46. seconds to run: 1.0.lurdruuulddrulldrrruulddruulldrdrdllurdrullluu

Puzzle: 86 F0EJKLIBAGDCNHMO

path length: 47. seconds to run: 1.03125.rrdldlulddruurrdlllurrurddllulddrruuulldrurdlul

Puzzle: 87 IEKJM0BAHFNLDGCO

path length: 54. seconds to run: 19.96875.drdluurdruuldldrruuldddlulurrdlldrurrululdlurrdlldruul

Puzzle: 88 OBLKNMIEACHG0JFD

path length: 65. seconds to run: 115.421875.rulurddluurrulldrdrulurdldrurdldruldllurrdllururrulldldrrruldluul

Puzzle: 89 KAGDJMCHIN0OFEBL

path length: 56. seconds to run: 6.921875.druulurdldlulurrdllddrrurulluldddruuldrrurdlullurdldruul

Puzzle: 90 EDGAKLNOJMHFB0IC

path length: 52. seconds to run: 7.140625.uuldrrdllurrurdlulddrurdluurdllururdldlulurrdldrullu

Puzzle: 91 IGEBNOLJKCFAHM0D

path length: 59. seconds to run: 5.96875.ulurrdlllurrrdllldrrrulldruluruldrurdldllurddlurulurdldruul

Puzzle: 92 CBGI0OLDFKENHMJA

path length: 57. seconds to run: 85.953125.ddrrruldluruldrurullldrrrdluullddrrdlluurrrulldrddruluull

Puzzle: 93 MINFLHABCD0GEJKO

path length: 50. seconds to run: 7.6875.llurrulddruldrdluluurrrdlurddldlulurdldruuulddrulu

Puzzle: 94 EGKH0NIMJLCOFADB

path length: 61. seconds to run: 1.640625.drdrrululddlurrurdluurdllurdlldrdrurullddruululdrdrulurddlluu

Puzzle: 95 DCFMGOI0JEHKBLAN

path length: 52. seconds to run: 9.4375.uldldldrururdluuldldrrdlurulurdldrdruuulldlurdrrulll

Puzzle: 96 AGONBFDILKMC0HEJ

path length: 51. seconds to run: 19.484375.rrulururddluulldrddlurdruuruldrddluululddrrruulldlu

Puzzle: 97 INEGHOABJDMFL0KC

path length: 46. seconds to run: 0.921875.uuurrdddluldrulurdllurrulldrurrdddluluurrdlull

Puzzle: 98 0KCLEBAIHJNOGDMF

path length: 56. seconds to run: 18.765625.rdrrulldddlurdrululurdlddrrrululdrurdldluurdluulddrdluuu

Puzzle: 99 GOD0JIBELKMFACNH

path length: 61. seconds to run: 15.109375.dddluluurdlddluuurrrdldrulldrulldrdrruulllurrrdlluldrdruuldlu

Puzzle: 100 KD0HFJEMLGNCABIO

path length: 56. seconds to run: 9.125.drdlululdrurrdllddrurullddlurdrurulluldrurdluldrdluurdlu

These puzzle solutions should generally be correct because the m value of 0.7 seemed to provide variability for puzzle lengths of around 50 – as shown from part 5. Because the solutions turned out to be around 50, 0.7 would be able to provide a certain amount of accuracy from its variability for the fastest possible speed. I ran the puzzles a couple times and each time the path lengths seemed to be correct, even with variability.

**Further Explorations:**

B) To implement this code, I set a counter outside of the while loop and incremented it every time I popped a node. After returning the path, I divided the counter value by the total time to run the program and this should return the nodes processed per second, accounting for even repeats in nodes processed. After running a few times I realized that ID-DFS nodes/sec were a little low. I then figured out that I needed to account for the nodes processed for the branches that did not result in the goal state. The correct results of the runs were:

|  |  |
| --- | --- |
| A\* | 22,500 nodes/sec |
| ID-DFS | 200,000 nodes/sec |
| BFS | 400,000 nodes/sec |

The results make sense because:

1. A\* uses a heap with a heuristic, and therefore looks through the minimal amount of nodes.
2. ID-DFS iterates through one path at a time (usually less amount of nodes than BFS), and takes a very long time, therefore the ratio of nodes to seconds is relatively small.
3. BFS takes relatively short periods of time to run, and traverses through every node up to the goal, and therefore has the highest nodes/second ration.

C) In order to implement the code to find the different puzzles, I used code from my original 8-Puzzle lab. Specifically, I augmented the code that finds the frequency of path lengths to be able to also store the states of each path length. I saved these in a dictionary of path lengths mapped to a list of states. I then randomly selected an index within the list to fulfill the requirement that the puzzles cannot be along the same solution. However, because I stored my states in a list – there are possible repeats – some of the puzzles would not appear as frequently as others.  
312045678  
312645078  
125340678  
312605748  
142370685  
014352678  
302614785  
325471680  
301752468  
158327640  
184320657  
084132657  
302716854  
318624057  
215470683  
042385167  
824130675  
425817360  
702635481  
041623857  
521847306  
750136284  
201748536  
734506281  
723854601  
785134260  
861047253  
678352041  
678540312  
236514870  
876041253

D) To generate the same list as above but for 24-puzzles, I had to do some further augmentation. In the code above, I would run through all the possibilities from the goal state, store the path lengths, then randomly select the puzzles. However, with 24-puzzles, it is impossible to run through every possible state before returning a list of selected puzzles. Therefore, I set an arbitrary breakpoint, say 30, and would keep building the dictionary until I got to path lengths of 30. This way, I would be able to return lists of puzzles depending on the time restraints I have (I could run it for 2-days or 20 mins). Here is the first 20:

EABCD0FGHIJKLMNOPQRSTUVWX

AB0CDEFGHIJKLMNOPQRSTUVWX

AFBCD0EGHIJKLMNOPQRSTUVWX

ABGCDEFLHIJK0MNOPQRSTUVWX

EABCDJFGHIKLQMNOP0RSTUVWX

ABGCDEFLHIJPKMNO0QRSTUVWX

AFB0DEKHCIJLGMNOPQRSTUVWX

AFBCDEKGHIJPLMNOQRSXTUVW0

EBCHDFAGMIJ0KLNOPQRSTUVWX

ABGCDEFHMIJPKLNOUQRS0TVWX

EABCDFKMGIJPH0NOQLRSTUVWX

AFBCDJEGHI0OLMNTKPRSUVQWX

EABCD0FKGIJPLHNOQVMSTUWRX

EABCDFGLHIOJKMSTPQNR0UVWX

ABGCI0ELDHJFQMNOKPRSTUVWX

EABCDJFLGHKPVMIOQR0NTUWXS

EABCDJGHRIKFM0NOLPVSTUWQX

EACH0JFBIDKLQMGOPRSNTUVWX

EABCDJFGHIPLVMNKOWUST0QRX

ABCDIJEFGHKLMNSTOPQR0UVWX