FOR RESULTS CONFIGURATIONS WERE SET MANUALLY.

1.

I chose to represent each object in hanoi as a class
This makes for an efficient data structure as no more than necessary space is used
And operations are simple and contained within their class.
Hashing a state is also simple because each object has its own hash function.

2.

I used an a_star approach (keeping a min heap) and evaluating the value of a state with a provided heuristic, the base heuristic being blind (returning 0) on any state.
I used two versions of each heuristic: consistent (with current depth) inconsistent (without)
A_star does not provide optimal results with inconsistent heuristics.

Bfs was provided as a flag within the astar function as it is a simple modification of the algorithm.

3.

k=number of disks

Heuristic: On_goal:(admissible) (consistent&inconsistent)
Returns k - number of correct disks on goal
Rationale: you need at least k - number of disks not on goal moves to solve the problem

Heuristic: on_goal_and_incorrect:(admissible) (consistent&inconsistent)
Returns:(k - number of correct disks on goal)+ 2*number of incorrect disks on goal (+depth)
Rationale: with an incorrect disk on the goal you require at least two moves to solve the problem for each incorrect disk (one to take it off, one to put it back on (correct location)).

Heuristic: blind (admissible) (consistent&inconsistent)
Returns 0 (+depth)

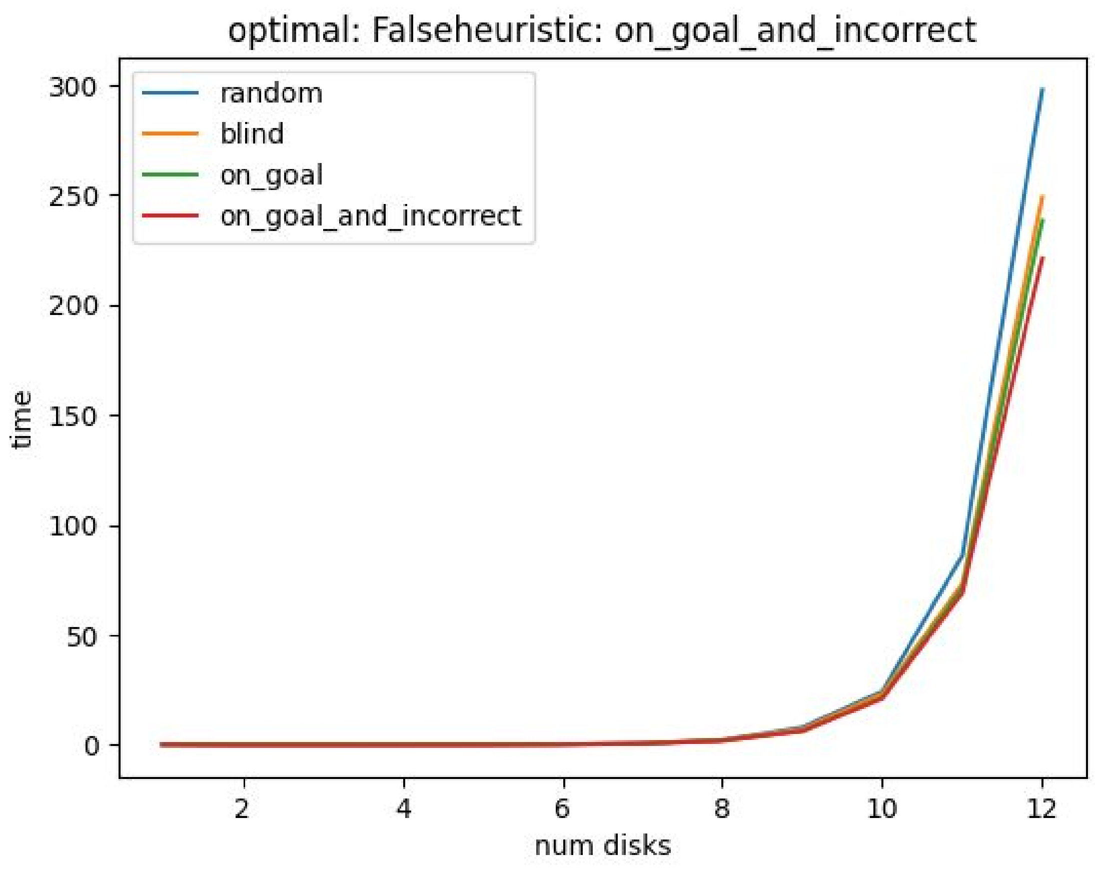
Heuristic: random (not admissible): (consistent&inconsistent)
Returns random int (1->10) (+depth)

4.

I used astar and bfs. Code is annotated, algorithm is well known and simple.
I used a min heap to keep track of min states given heuristic.
For bfs I modified it so new expanded states would only be added once all previous ones were explored.

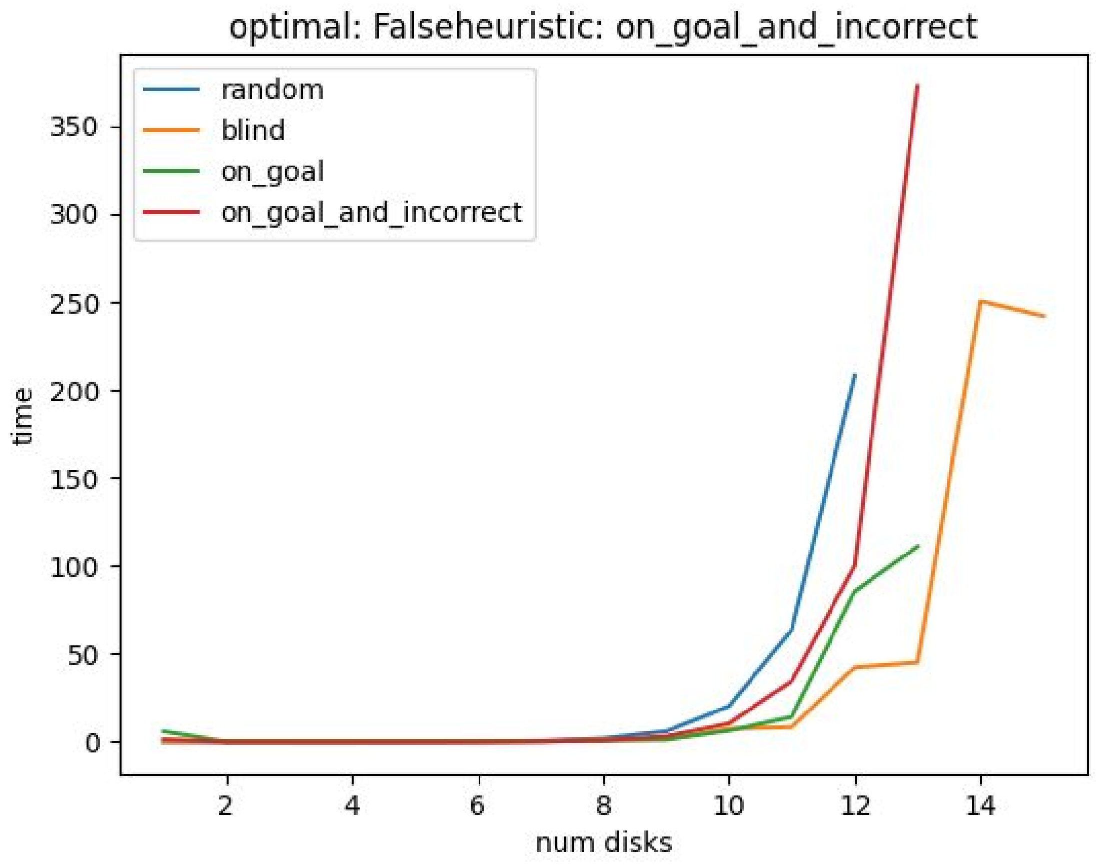
5.
Max m for 10 min ranged for 12 ->15 depending on heuristic (consistent/inconsistend) see 10 min folders for detailed results graphs are bellow:
Solutions for astar are optimal if heuristic was consistent and admissible
Otherwise no guarantee for optimality

A* 10 min consistent
Consistent:



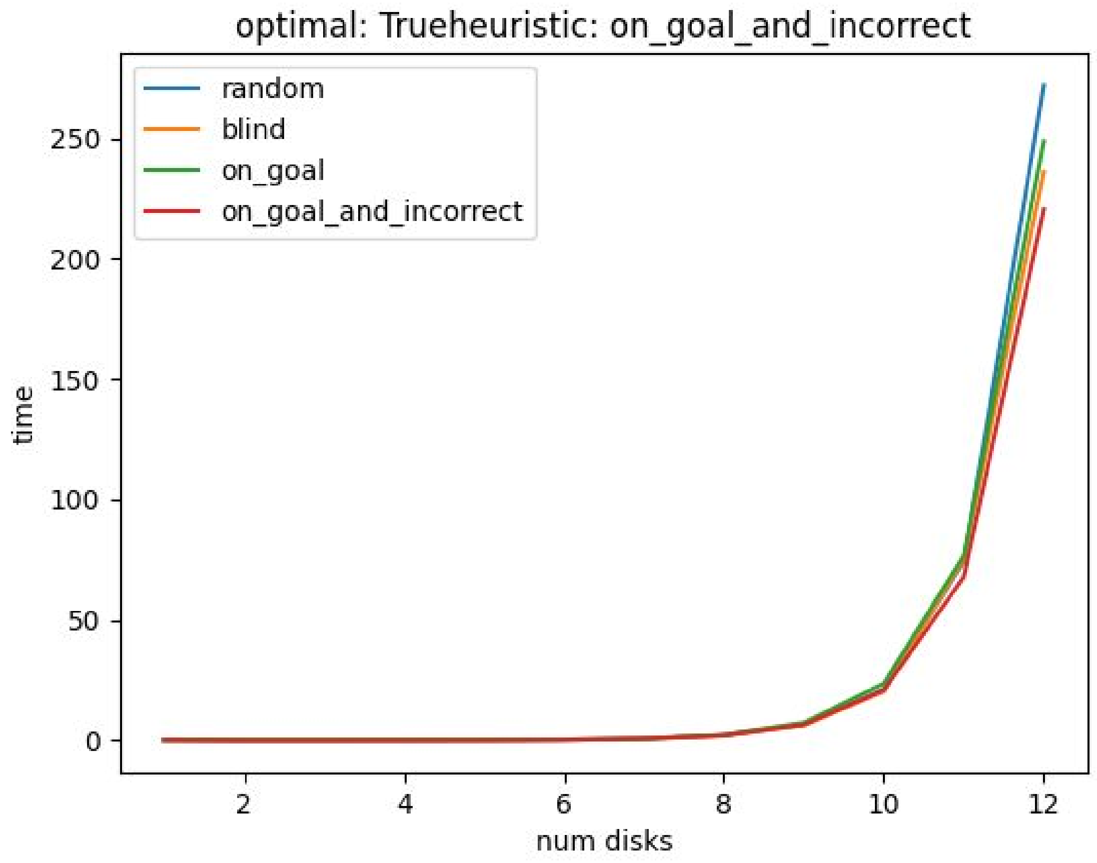
SEE Log file for plan lengths
All runs provided optimal path except for random as that is not admissible

A* inconsistent 10 minutes max eval



If line has stopped it means that the next evaluation timed out (10 min)

Pattern Database (10 minutes max)
Note time difference is due to complexity of heuristic and background processes running at the time of evaluation.



Maximum number of disks found optimally is 12 (10 min max)
This can be seen in the astar consistent or bfs folders (for the plans)
Here is the sample log for the on_goal_and_incorrect heuristic run on PDB_bfs…

```
on_goal_and_incorrect
visited | plan_length | time(s)
4 | 1 | 0.16076183319091797
10 | 3 | 0.006538867950439453
28 | 7 | 0.00953817367553711
82 | 15 | 0.04976058006286621
244 | 31 | 0.09459924697875977
730 | 63 | 0.19930124282836914
2188 | 127 | 0.6568489074707031
6562 | 255 | 1.973480463027954
19684 | 511 | 6.405580997467041
59050 | 1023 | 20.81358289718628
177148 | 2047 | 67.76588940620422
531442 | 4095 | 220.44842314720154
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
```

Maximum number of disks for plan found sub-optimally (10 min max) is 15 using blind inconsistent heuristic on a* log is here:

```
blind
visited | plan_length | time(s)
3 | 1 | 0.1516413688659668
6 | 3 | 0.004004240036010742
21 | 7 | 0.004143476486206055
62 | 16 | 0.013217926025390625
198 | 39 | 0.05263328552246094
530 | 89 | 0.12938857078552246
898 | 135 | 0.24203062057495117
3755 | 423 | 1.0645358562469482
4715 | 519 | 1.4296157360076904
22548 | 1939 | 7.497288465499878
24136 | 2055 | 8.454140424728394
114965 | 8155 | 42.358295917510986
116515 | 8199 | 45.1737277507782
618917 | 32711 | 250.49558424949646
541719 | 32775 | 242.16327905654907
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

For plan see the relevant directory

For pdb less than 5 min the max number of disks is still 12 as seen in the log files/graphs which were run for a timeout of 10min.