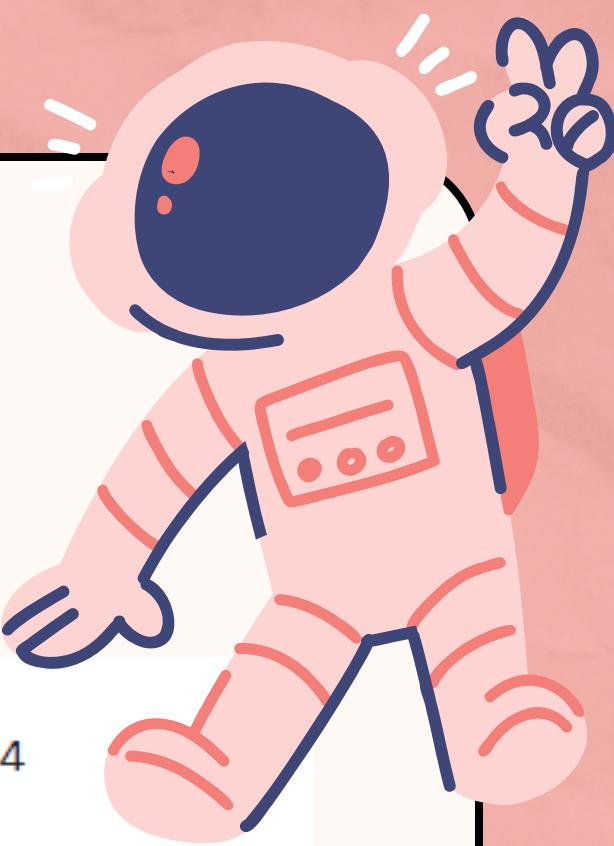


8 puzzle

COMPARING OUR ALGORITHMS



Simple Example



```
7
[[ '3' '5' '1']
 ['4' ' ' '2']
 ['6' '7' '8']]
[[ ' ' '3' '1']
 ['4' '5' '2']
 ['6' '7' '8']]
8
[[ ' ' '3' '2']
 ['4' '1' '5']
 ['6' '7' '8']]
[[ '3' '2' ' ']
 ['4' '1' '5']
 ['6' '7' '8']]
9
[[ '3' '1' '2']
 ['4' '7' '5']
 [' ' '6' '8']]
[[ '3' '1' '2']
 [' ' '4' '5']
 ['6' '7' '8']]
10
We have reached the goal node!
[[ ' ' '1' '2']
 ['3' '4' '5']
 ['6' '7' '8']]
```

BFS

10

A* hamming 3

```
2
g-score = 3 --- h_score = 4
[['3' ' ' '2']
 ['4' '1' '5']
 ['6' '7' '8']]
g-score = 3 --- h_score = 4
[['3' '1' '2']
 ['4' '7' '5']
 ['6' ' ' '8']]
g-score = 3 --- h_score = 2
[['3' '1' '2']
 [' ' '4' '5']
 ['6' '7' '8']]
3
We have reached the goal node!
g-score = 4 --- h_score = 0
[[' ' '1' '2']
 ['3' '4' '5']
 ['6' '7' '8']]
```

```
2
g-score = 3 --- h_score = 4
[['3' ' ' '2']
 ['4' '1' '5']
 ['6' '7' '8']]
g-score = 3 --- h_score = 6
[['3' '1' '2']
 ['4' '7' '5']
 ['6' ' ' '8']]
g-score = 3 --- h_score = 2
[['3' '1' '2']
 [' ' '4' '5']
 ['6' '7' '8']]
3
We have reached the goal node!
g-score = 4 --- h_score = 0
[[' ' '1' '2']
 ['3' '4' '5']
 ['6' '7' '8']]
g-score = 4 --- h_score = 4
```

A* manhattan 3

Simple Example

```
2
h_score = 4
[['3' ' ' '2'],
 ['4' '1' '5'],
 ['6' '7' '8']]
h_score = 4
[['3' '1' '2'],
 ['4' '7' '5'],
 ['6' ' ' '8']]
h_score = 2
[['3' '1' '2'],
 [' ' '4' '5'],
 ['6' '7' '8']]
3
We have reached the goal node!
```

```
h_score = 0
[[' ' '1' '2'],
 ['3' '4' '5'],
 ['6' '7' '8']]
```

greedy hamming

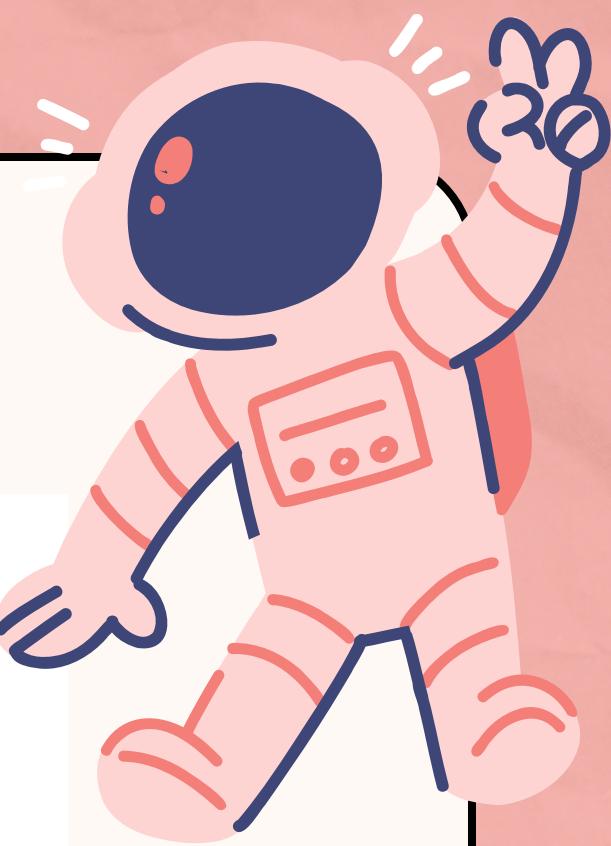
3

```
2
h_score = 4
[['3' ' ' '2'],
 ['4' '1' '5'],
 ['6' '7' '8']]
h_score = 6
[['3' '1' '2'],
 ['4' '7' '5'],
 ['6' ' ' '8']]
h_score = 2
[['3' '1' '2'],
 [' ' '4' '5'],
 ['6' '7' '8']]
3
We have reached the goal node!
```

```
h_score = 0
[[' ' '1' '2'],
 ['3' '4' '5'],
 ['6' '7' '8']]
```

greedy manhattan

3



Simple Example

```
10  
We have reached the goal node!  
[[ '_' '1' '2'  
  ['3' '4' '5'  
  ['6' '7' '8']]
```

BFS 10

```
3  
We have reached the goal node!  
g-score = 4 --- h_score = 0  
[[ '_' '1' '2'  
  ['3' '4' '5'  
  ['6' '7' '8']]
```

A* hamming 3

```
3  
We have reached the goal node!  
g-score = 4 --- h_score = 0  
[[ '_' '1' '2'  
  ['3' '4' '5'  
  ['6' '7' '8']]  
g-score = 4 --- h score = 4
```

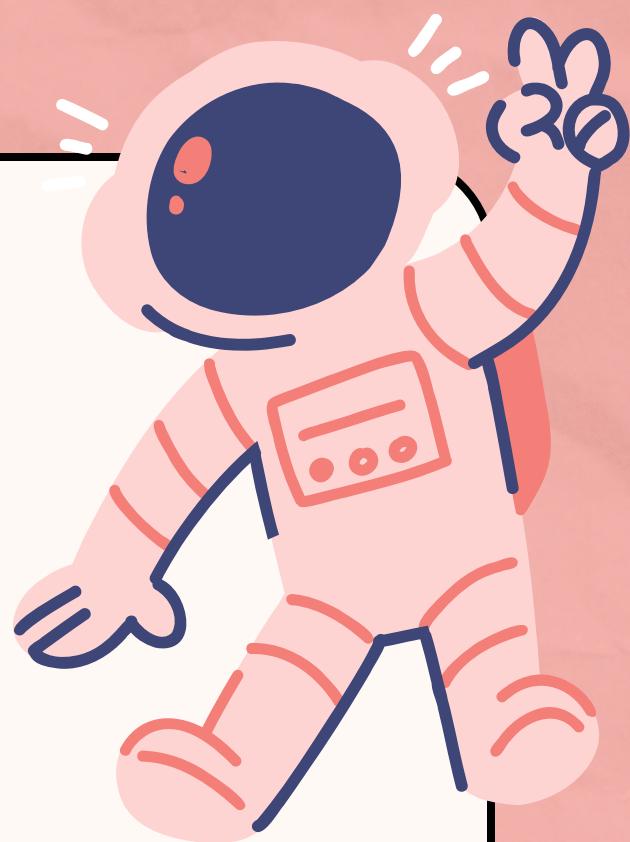
A* manhattan 3

```
3  
We have reached the goal node!  
h_score = 0  
[[ '_' '1' '2'  
  ['3' '4' '5'  
  ['6' '7' '8']]
```

greedy hamming 3

```
3  
We have reached the goal node!  
h_score = 0  
[[ '_' '1' '2'  
  ['3' '4' '5'  
  ['6' '7' '8']]
```

greedy manhattan 3



Hard Example

```
19997  
[[8 4 _]  
 [7 2 6]  
 [1 5 3]]  
[[8 4 6]  
 [7 2 3]  
 [1 5 _]]  
  
19998  
[[4 2 7]  
 [8 _ 3]  
 [1 6 5]]  
[[4 2 7]  
 [8 6 3]  
 [_ 1 5]]  
  
19999  
[[4 2 7]  
 [8 5 6]  
 [_ 1 3]]  
[[4 2 7]  
 [8 5 6]  
 [1 3 _]]
```

BFS

19999...

```
6522  
g-score = 20 --- h_score = 4  
[[3 _ 2]  
 [4 1 5]  
 [6 7 8]]  
g-score = 20 --- h_score = 2  
[[3 1 2]  
 [_ 4 5]  
 [6 7 8]]  
g-score = 20 --- h_score = 4  
[[3 1 2]  
 [4 5 _]  
 [6 7 8]]  
6523  
g-score = 21 --- h_score = 3  
[[1 4 2]  
 [3 _ 5]  
 [6 7 8]]  
We have reached the goal node!  
g-score = 21 --- h_score = 0  
[[_ 1 2]  
 [3 4 5]  
 [6 7 8]]
```

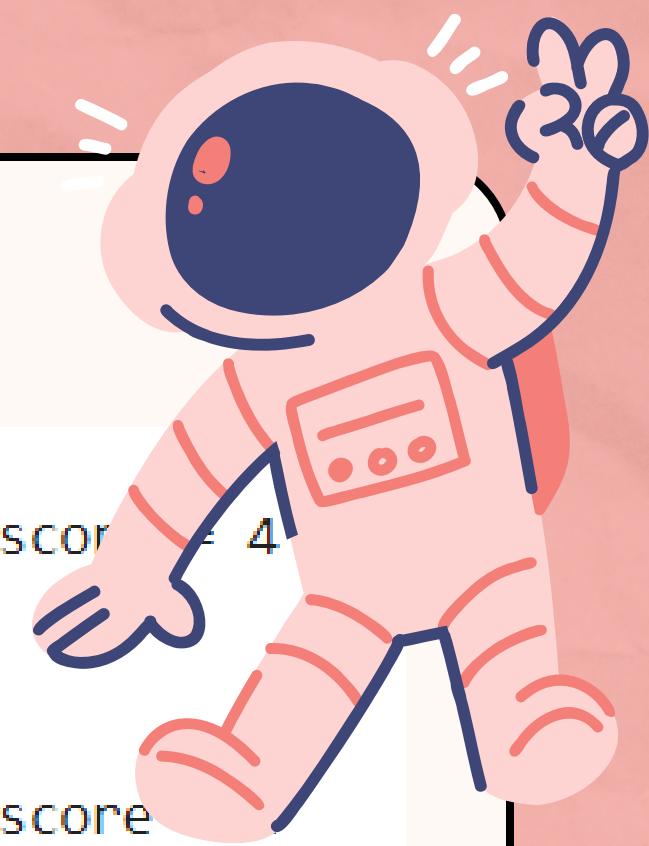
A* hamming

6523

```
1741  
g-score = 20 --- h_score = 4  
[[3 _ 2]  
 [4 1 5]  
 [6 7 8]]  
g-score = 20 --- h_score = 6  
[[3 1 2]  
 [4 5 _]  
 [6 7 8]]  
1742  
We have reached the goal node!  
g-score = 21 --- h_score = 0  
[[_ 1 2]  
 [3 4 5]  
 [6 7 8]]
```

A* manhattan

1742



Hard Example

```
785
h_score = 5
[['3' '1' '_'],
 ['4' '5' '2'],
 ['6' '7' '8']]
h_score = 3
[['3' '1' '2'],
 ['4' '_' '5'],
 ['6' '7' '8']]
786
h_score = 4
[['3' '_' '2'],
 ['4' '1' '5'],
 ['6' '7' '8']]
h_score = 4
[['3' '1' '2'],
 ['4' '7' '5'],
 ['6' '_' '8']]
h_score = 2
[['3' '1' '2'],
 ['_' '4' '5'],
 ['6' '7' '8']]
787
We have reached the goal node!
h_score = 0
[['_' '1' '2'],
 ['3' '4' '5'],
 ['6' '7' '8']]
h_score = 0
```

greedy hamming

787

```
133
h_score = 4
[['3' '1' '2'],
 ['6' '4' '5'],
 ['_' '7' '8']]
h_score = 8
[['3' '1' '2'],
 ['6' '4' '5'],
 ['7' '8' '_']]
134
h_score = 2
[['3' '1' '2'],
 ['_' '4' '5'],
 ['6' '7' '8']]
135
We have reached the goal node!
h_score = 0
[['_' '1' '2'],
 ['3' '4' '5'],
 ['6' '7' '8']]
```

greedy manhattan

135



Hard Example

```
19999  
[['4' '2' '7'],  
 ['8' '5' '6'],  
 ['-' '1' '3'],  
 [['4' '2' '7'],  
 ['8' '5' '6'],  
 ['1' '3' '-']]
```

BFS

19999...

NA

```
6523  
g-score = 21 --- h_score = 3  
[['1' '4' '2'],  
 ['3' '-' '5'],  
 ['6' '7' '8']]
```

We have reached the goal node!

```
g-score = 21 --- h_score = 0  
[['-' '1' '2'],  
 ['3' '4' '5'],  
 ['6' '7' '8']]
```

A* hamming

6523

21

```
1742  
We have reached the goal node!  
g-score = 21 --- h_score = 0  
[['-' '1' '2'],  
 ['3' '4' '5'],  
 ['6' '7' '8']]
```

A* manhattan

1742

21

```
787  
We have reached the goal node!  
h_score = 0  
[['-' '1' '2'],  
 ['3' '4' '5'],  
 ['6' '7' '8']]
```

greedy hamming

787

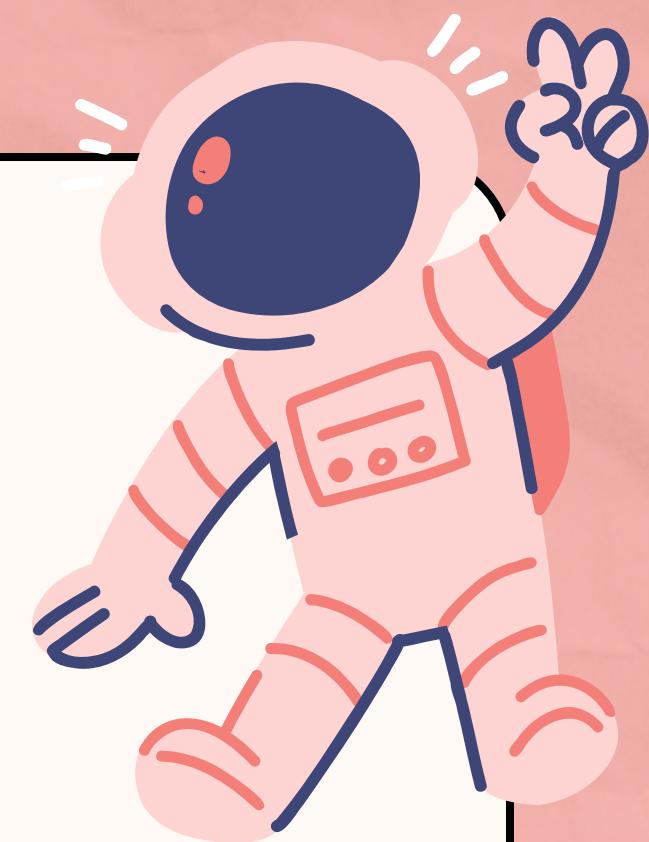
45

```
135  
We have reached the goal node!  
h_score = 0  
[['-' '1' '2'],  
 ['3' '4' '5'],  
 ['6' '7' '8']]
```

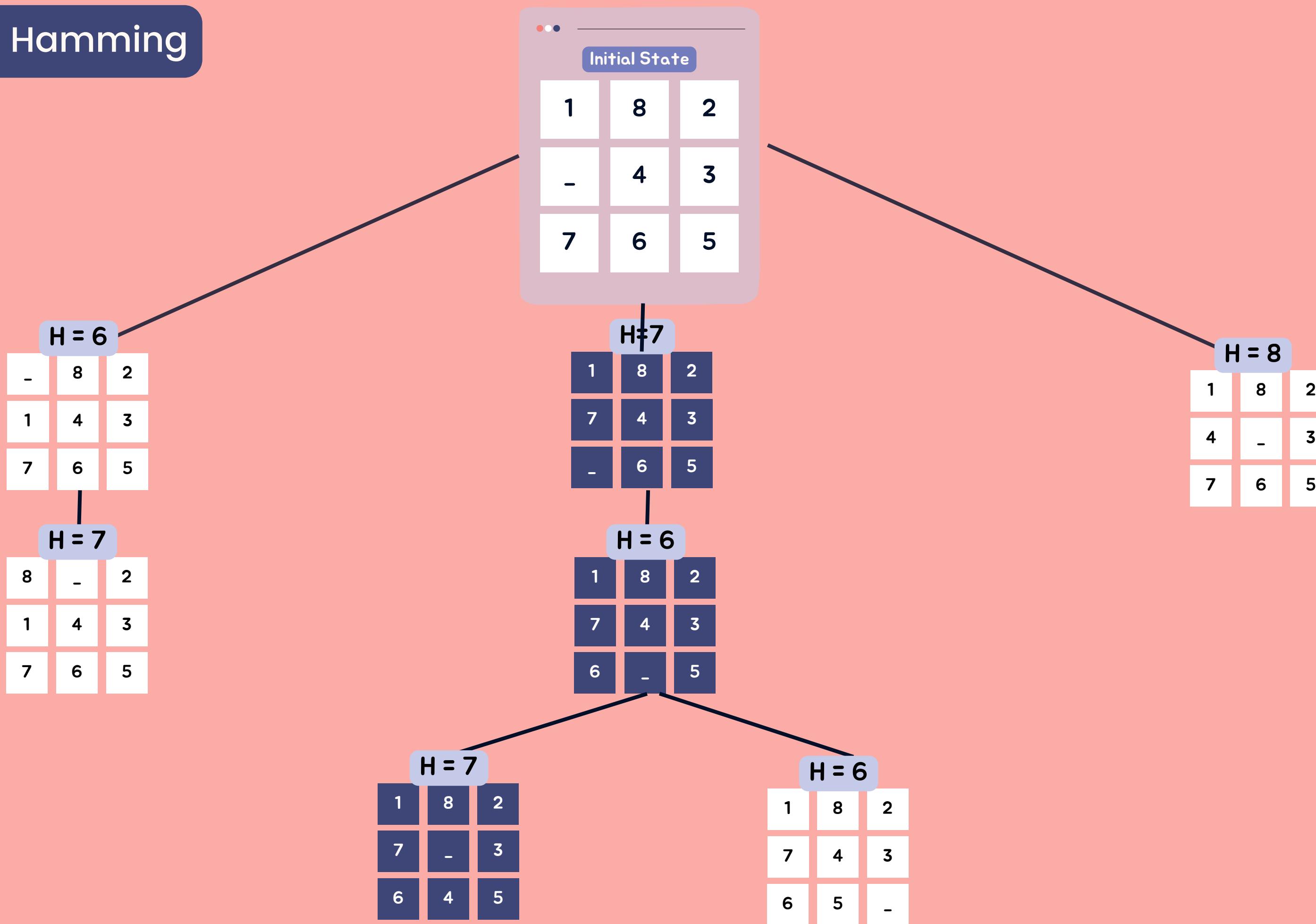
greedy manhattan

135

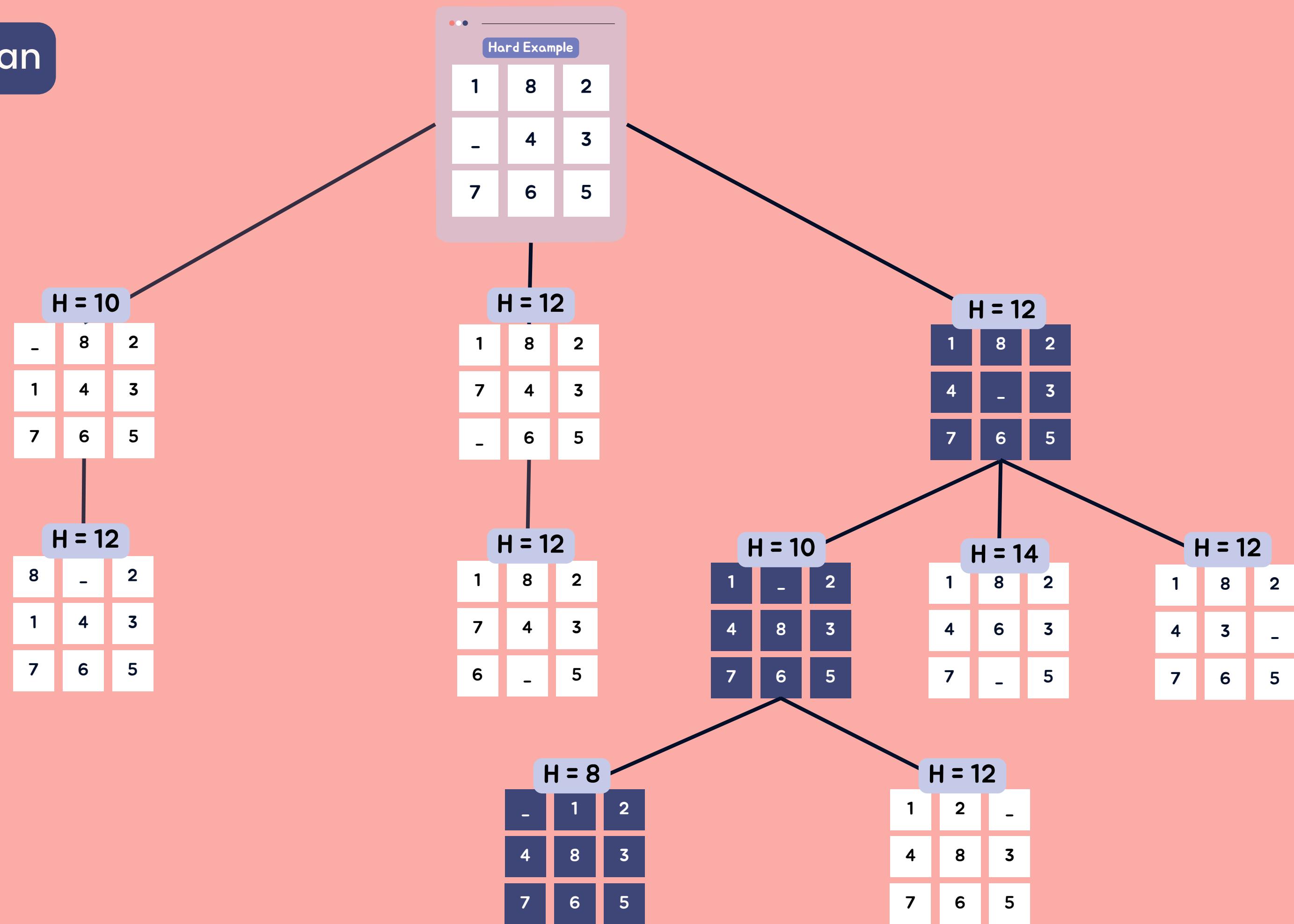
31



Best First: Hamming



Best First: Manhattan



Best-first: Manhattan

Continuing the path from where we left off...

Hard Example

1	8	2
-	4	3
7	6	5

Move 3

-	1	2
4	8	3
7	6	5

Move 4

4	1	2
-	8	3
7	6	5

Move 5

4	1	2
7	8	3
-	6	5

Move 6

4	1	2
7	8	3
6	-	5

Move 7

4	1	2
7	-	3
6	8	5

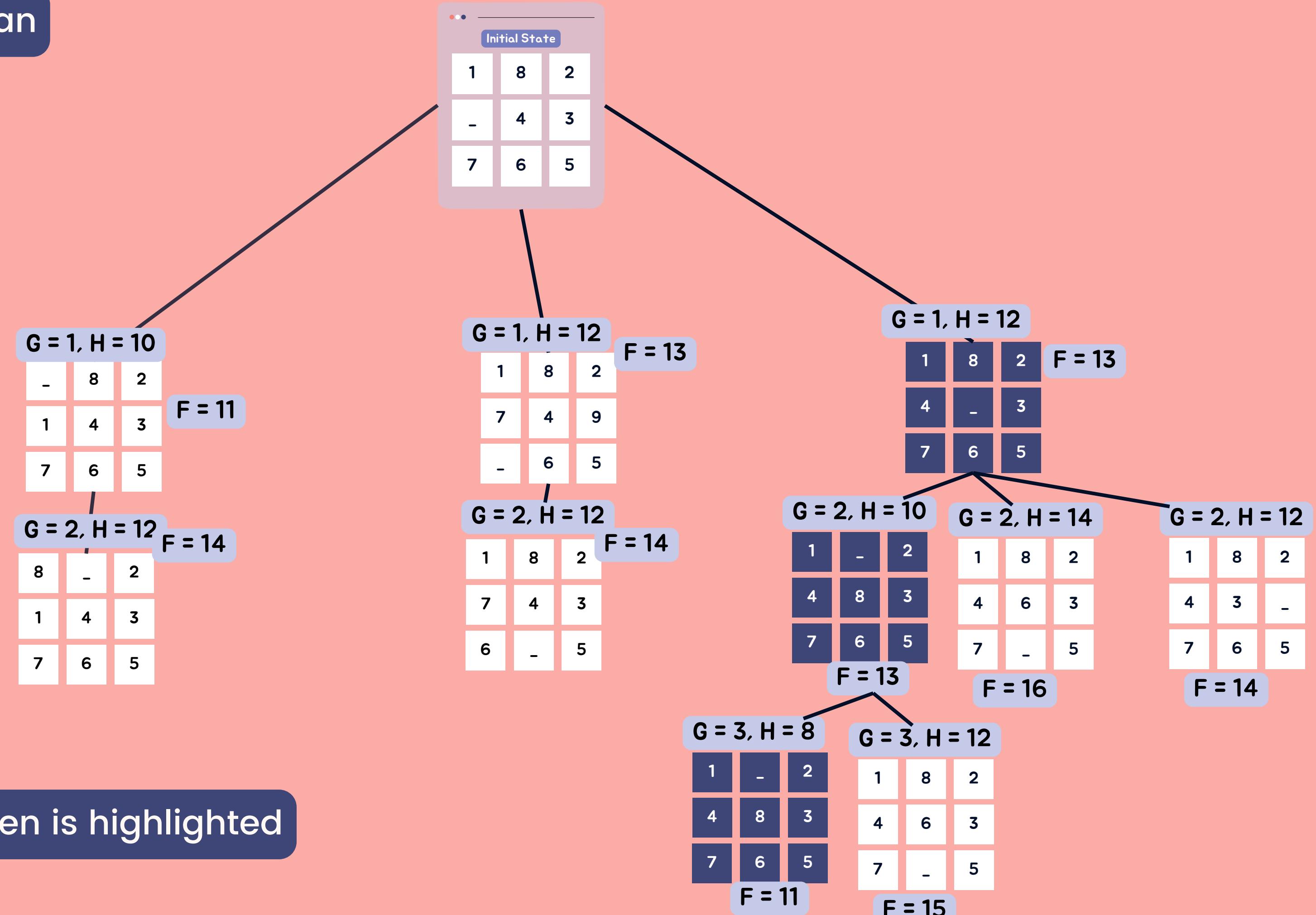
Move 8

4	1	2
7	3	-
6	8	5

Move 9

4	1	2
7	3	5
6	8	-

A*: Manhattan



The path chosen is highlighted

A*: Manhattan

Continuing the path from where we left off...

Hard Example

1	8	2
-	4	3
7	6	5

Move 3

-	1	2
4	8	3
7	6	5

Move 4

4	1	2
-	8	3
7	6	5

Move 5

4	1	2
7	8	3
-	6	5

Move 6

4	1	2
7	8	3
6	-	5

Move 7

4	1	2
7	-	3
6	8	5

Move 8

4	1	-
7	3	-
6	8	5

Move 9

4	1	-
7	3	2
6	8	5

Hard Example

Correct Path

Initial State

1	8	2
—	4	3
7	6	5

Move number: 1

1	8	2
7	4	3
—	6	5

Move number: 2

1	8	2
7	4	3
6	—	5

Move number: 3

1	8	2
7	—	3
6	4	5

Move number: 4

1	—	2
7	8	3
6	4	5

greedy hamming

Correct Path

Initial State

1	8	2
—	4	3
7	6	5

Move number: 1

1	8	2
4	—	3
7	6	5

Move number: 2

1	—	2
4	8	3
7	6	5

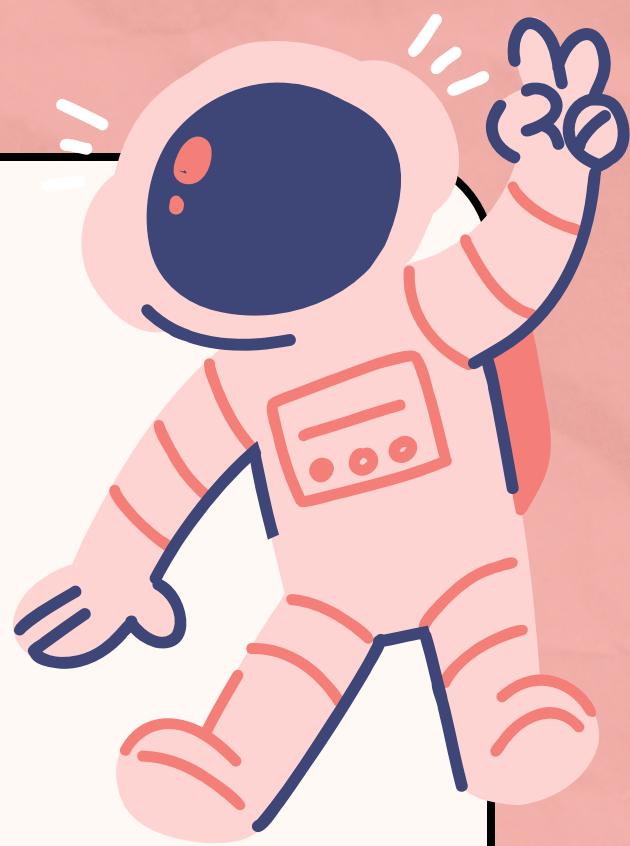
Move number: 3

—	1	2
4	8	3
7	6	5

Move number: 4

4	1	2
—	8	3
7	6	5

greedy manhattan





IN CONCLUSION



WE HAVE SHOWN THAT BFS TAKES LONGER AND GOES THROUGH MORE UNNECESSARY STEPS COMPARED TO BEST FIRST AND A* SEARCH. IN THE SIMPLE EXAMPLE, IT SEEMS THAT ALL OF OUR ALGORITHMS HAVE THE SAME PATHS/HAS THE SAME SEQUENCE OF NODES TO GET TO THE GOAL STATE

WHEN USING THE HARD EXAMPLE, WE SAW THAT STILL, THE BFS IS THE WORST AS IT DIDNT EVEN GET TO FIND THE GOAL IN 20K NODE EXPANSIONS WHILE THE REST WERE ABLE TO. GREEDY BEST FIRST SEARCH TOOK MUCH LESS TIME FINDING THE GOAL NODE; HOWEVER, A* SEARCH WAS ABLE TO FIND THE MORE OPTIMAL PATH.

IT WAS ALSO SHOWN THAT USING MANHATTAN DISTANCE WAS MUCH MORE OPTIMAL COMPARED TO USING HAMMING DISTANCE.





thank you for

LISTENING!

yay!!!