# PANCAKE SORTING

**Fatih Alperen Acarel**
**160316055**

## 1.Problem Formulation

Firstly I take number of pancakes from user and according to this number I add numbers begining from 0 until to number of pancakes to the **GOAL** and **action_list** tuples.After that user have choice for determine the initial state.If user want to ordering by own code will fill the "**initia**l" tuple with users ordering.If user do not wants to ordering by own then the tuple will fill automatically and code will  select a tuple randomly from permutations of **GOAL** tuple.

After User's process completed, our model comes into play.Firstly model select a action from **action_list** and the result function returns the new state.The new state calculated as follows;

I take the reverse of initial tuple until integer of action and then I add the rest of all tuple.After that **is_goal** function controls, is **initial** tuple equal to the **GOAL** tuple? Until reaching to the goal We get one point penalty per step.

## 2.Heuristic Function

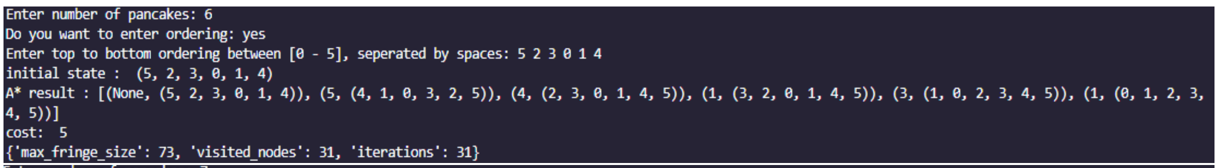
In Heuristic Function I compare the last two elements of **initial** tuple and the **GOAL** tuple.

## 3.Discussion on the Results

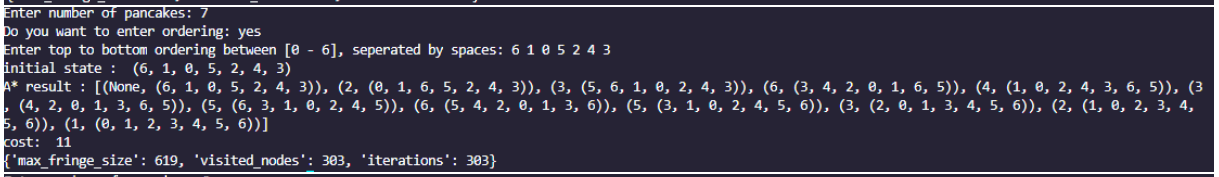
Except Limited Deep First and Iterative Deepening Search**;** Informed search algorithms found solution more quickly than uninformed search.Also Informed Search's efficieny was highly and cost was less than Uninformed Search.Sometimes Uninformed Search's cannot reach the result.

## Test

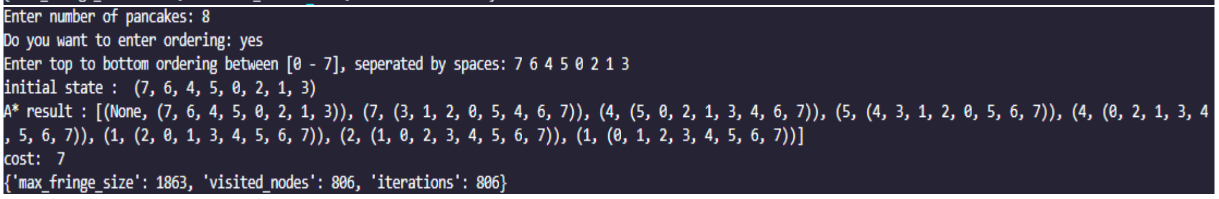
## A*:



```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
A* result : [(None, (5, 2, 3, 0, 1, 4)), (5, (4, 1, 0, 3, 2, 5)), (4, (2, 3, 0, 1, 4, 5)), (1, (3, 2, 0, 1, 4, 5)), (3, (1, 0, 2, 3, 4, 5)), (1, (0, 1, 2, 3,
4, 5))]
cost:  5
{'max_fringe_size': 73, 'visited_nodes': 31, 'iterations': 31}
```

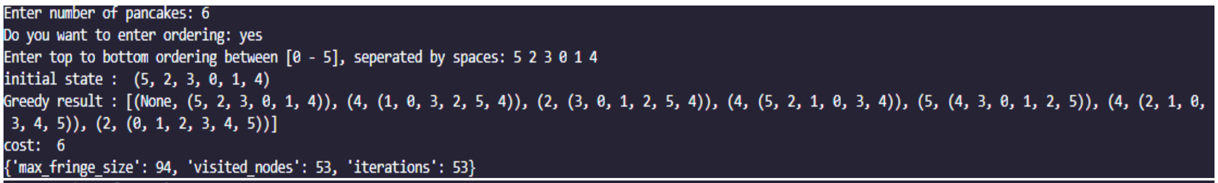


```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
A* result : [(None, (6, 1, 0, 5, 2, 4, 3)), (2, (0, 1, 6, 5, 2, 4, 3)), (3, (5, 6, 1, 0, 2, 4, 3)), (6, (3, 4, 2, 0, 1, 6, 5)), (4, (1, 0, 2, 4, 3, 6, 5)), (3
, (4, 2, 0, 1, 3, 6, 5)), (5, (6, 3, 1, 0, 2, 4, 5)), (6, (5, 4, 2, 0, 1, 3, 6)), (5, (3, 1, 0, 2, 4, 5, 6)), (3, (2, 0, 1, 3, 4, 5, 6)), (2, (1, 0, 2, 3, 4,
5, 6)), (1, (0, 1, 2, 3, 4, 5, 6))]
cost:  11
{'max_fringe_size': 619, 'visited_nodes': 303, 'iterations': 303}
```

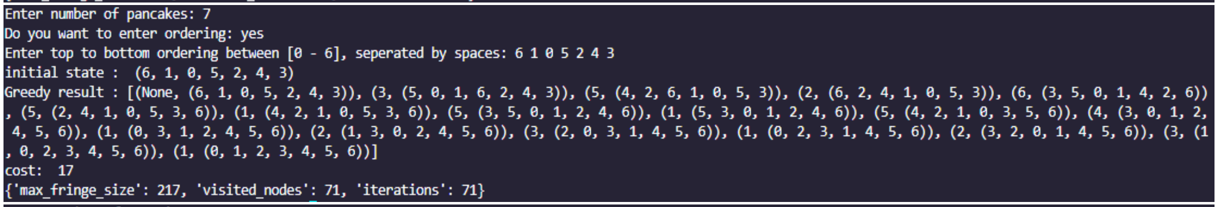


```
Enter number of pancakes: 8
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 7], seperated by spaces: 7 6 4 5 0 2 1 3
initial state :  (7, 6, 4, 5, 0, 2, 1, 3)
A* result : [(None, (7, 6, 4, 5, 0, 2, 1, 3)), (7, (3, 1, 2, 0, 5, 4, 6, 7)), (4, (5, 0, 2, 1, 3, 4, 6, 7)), (5, (4, 3, 1, 2, 0, 5, 6, 7)), (4, (0, 2, 1, 3, 4
, 5, 6, 7)), (1, (2, 0, 1, 3, 4, 5, 6, 7)), (2, (1, 0, 2, 3, 4, 5, 6, 7)), (1, (0, 1, 2, 3, 4, 5, 6, 7))]
cost:  7
{'max_fringe_size': 1863, 'visited_nodes': 806, 'iterations': 806}
```

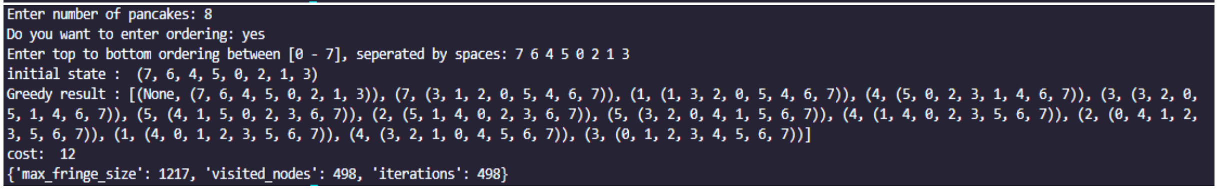
## Greedy:



```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
Greedy result : [(None, (5, 2, 3, 0, 1, 4)), (4, (1, 0, 3, 2, 5, 4)), (2, (3, 0, 1, 2, 5, 4)), (4, (5, 2, 1, 0, 3, 4)), (5, (4, 3, 0, 1, 2, 5)), (4, (2, 1, 0,
 3, 4, 5)), (2, (0, 1, 2, 3, 4, 5))]
cost:  6
{'max_fringe_size': 94, 'visited_nodes': 53, 'iterations': 53}
```

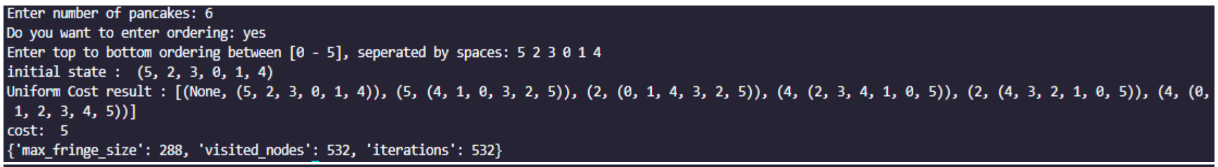


```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
Greedy result : [(None, (6, 1, 0, 5, 2, 4, 3)), (3, (5, 0, 1, 6, 2, 4, 3)), (5, (4, 2, 6, 1, 0, 5, 3)), (2, (6, 2, 4, 1, 0, 5, 3)), (6, (3, 5, 0, 1, 4, 2, 6))
, (5, (2, 4, 1, 0, 5, 3, 6)), (1, (4, 2, 1, 0, 5, 3, 6)), (5, (3, 5, 0, 1, 2, 4, 6)), (1, (5, 3, 0, 1, 2, 4, 6)), (5, (4, 2, 1, 0, 3, 5, 6)), (4, (3, 0, 1, 2,
 4, 5, 6)), (1, (0, 3, 1, 2, 4, 5, 6)), (2, (1, 3, 0, 2, 4, 5, 6)), (3, (2, 0, 3, 1, 4, 5, 6)), (1, (0, 2, 3, 1, 4, 5, 6)), (2, (3, 2, 0, 1, 4, 5, 6)), (3, (1
, 0, 2, 3, 4, 5, 6)), (1, (0, 1, 2, 3, 4, 5, 6))]
cost:  17
{'max_fringe_size': 217, 'visited_nodes': 71, 'iterations': 71}
```

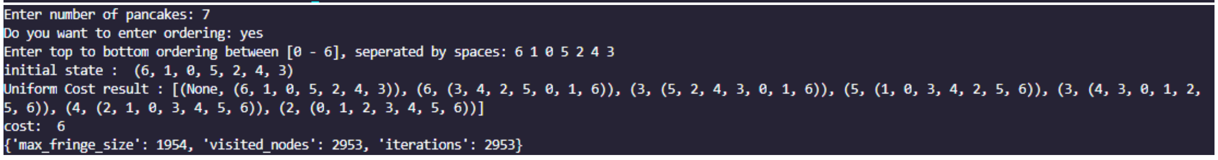


```
Enter number of pancakes: 8
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 7], seperated by spaces: 7 6 4 5 0 2 1 3
initial state :  (7, 6, 4, 5, 0, 2, 1, 3)
Greedy result : [(None, (7, 6, 4, 5, 0, 2, 1, 3)), (7, (3, 1, 2, 0, 5, 4, 6, 7)), (1, (1, 3, 2, 0, 5, 4, 6, 7)), (4, (5, 0, 2, 3, 1, 4, 6, 7)), (3, (3, 2, 0,
5, 1, 4, 6, 7)), (5, (4, 1, 5, 0, 2, 3, 6, 7)), (2, (5, 1, 4, 0, 2, 3, 6, 7)), (5, (3, 2, 0, 4, 1, 5, 6, 7)), (4, (1, 4, 0, 2, 3, 5, 6, 7)), (2, (0, 4, 1, 2,
3, 5, 6, 7)), (1, (4, 0, 1, 2, 3, 5, 6, 7)), (4, (3, 2, 1, 0, 4, 5, 6, 7)), (3, (0, 1, 2, 3, 4, 5, 6, 7))]
cost:  12
{'max_fringe_size': 1217, 'visited_nodes': 498, 'iterations': 498}
```

## Uniform Cost:



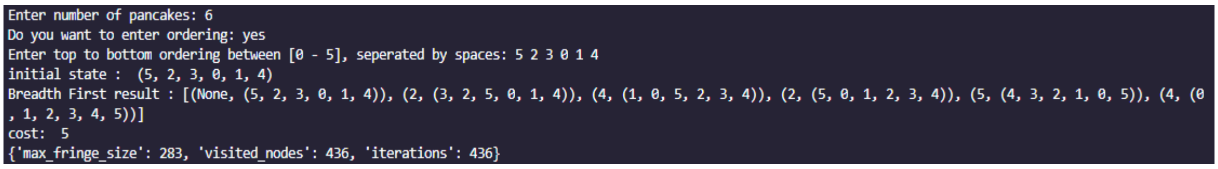
```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
Uniform Cost result : [(None, (5, 2, 3, 0, 1, 4)), (5, (4, 1, 0, 3, 2, 5)), (2, (0, 1, 4, 3, 2, 5)), (4, (2, 3, 4, 1, 0, 5)), (2, (4, 3, 2, 1, 0, 5)), (4, (0,
 1, 2, 3, 4, 5))]
cost:  5
{'max_fringe_size': 288, 'visited_nodes': 532, 'iterations': 532}
```



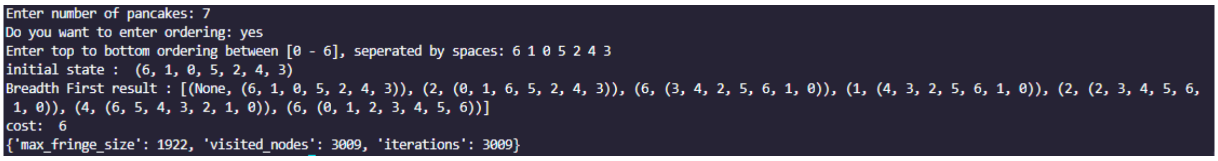
```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
Uniform Cost result : [(None, (6, 1, 0, 5, 2, 4, 3)), (6, (3, 4, 2, 5, 0, 1, 6)), (3, (5, 2, 4, 3, 0, 1, 6)), (5, (1, 0, 3, 4, 2, 5, 6)), (3, (4, 3, 0, 1, 2,
5, 6)), (4, (2, 1, 0, 3, 4, 5, 6)), (2, (0, 1, 2, 3, 4, 5, 6))]
cost:  6
{'max_fringe_size': 1954, 'visited_nodes': 2953, 'iterations': 2953}
```

*(It cannot reached to the result for 7 6 4 5 0 2 1 3 )*

## Breadth First:



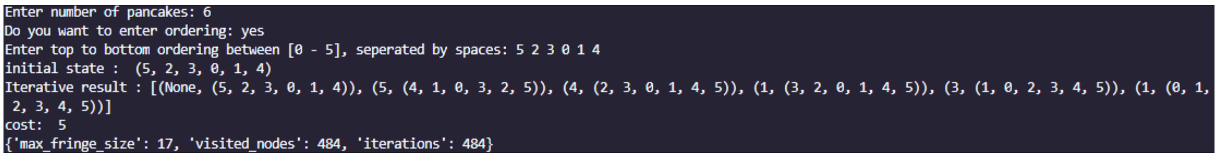
```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
Breadth First result : [(None, (5, 2, 3, 0, 1, 4)), (2, (3, 2, 5, 0, 1, 4)), (4, (1, 0, 5, 2, 3, 4)), (2, (5, 0, 1, 2, 3, 4)), (5, (4, 3, 2, 1, 0, 5)), (4, (0
, 1, 2, 3, 4, 5))]
cost:  5
{'max_fringe_size': 283, 'visited_nodes': 436, 'iterations': 436}
```



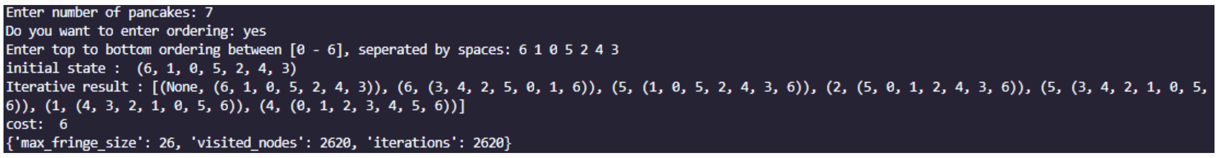
```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
Breadth First result : [(None, (6, 1, 0, 5, 2, 4, 3)), (2, (0, 1, 6, 5, 2, 4, 3)), (6, (3, 4, 2, 5, 6, 1, 0)), (1, (4, 3, 2, 5, 6, 1, 0)), (2, (2, 3, 4, 5, 6,
 1, 0)), (4, (6, 5, 4, 3, 2, 1, 0)), (6, (0, 1, 2, 3, 4, 5, 6))]
cost:  6
{'max_fringe_size': 1922, 'visited_nodes': 3009, 'iterations': 3009}
```

*(It cannot reached to the result for 7 6 4 5 0 2 1 3 )*

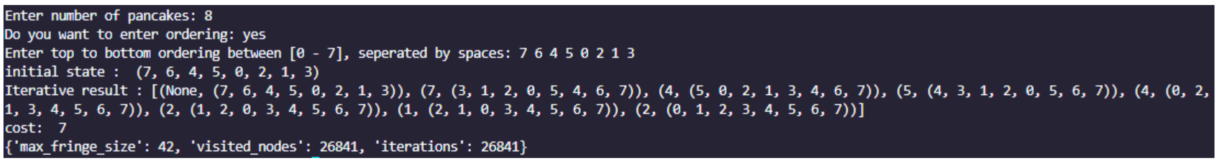
## Iterative Deepening Search:



```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
Iterative result : [(None, (5, 2, 3, 0, 1, 4)), (5, (4, 1, 0, 3, 2, 5)), (4, (2, 3, 0, 1, 4, 5)), (1, (3, 2, 0, 1, 4, 5)), (3, (1, 0, 2, 3, 4, 5)), (1, (0, 1,
 2, 3, 4, 5))]
cost:  5
{'max_fringe_size': 17, 'visited_nodes': 484, 'iterations': 484}
```

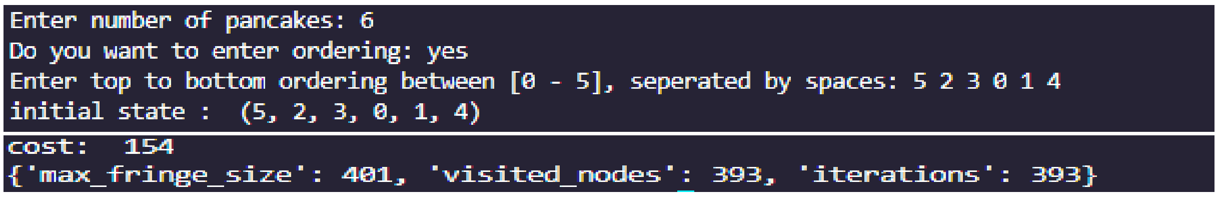


```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
Iterative result : [(None, (6, 1, 0, 5, 2, 4, 3)), (6, (3, 4, 2, 5, 0, 1, 6)), (5, (1, 0, 5, 2, 4, 3, 6)), (2, (5, 0, 1, 2, 4, 3, 6)), (5, (3, 4, 2, 1, 0, 5,
6)), (1, (4, 3, 2, 1, 0, 5, 6)), (4, (0, 1, 2, 3, 4, 5, 6))]
cost:  6
{'max_fringe_size': 26, 'visited_nodes': 2620, 'iterations': 2620}
```

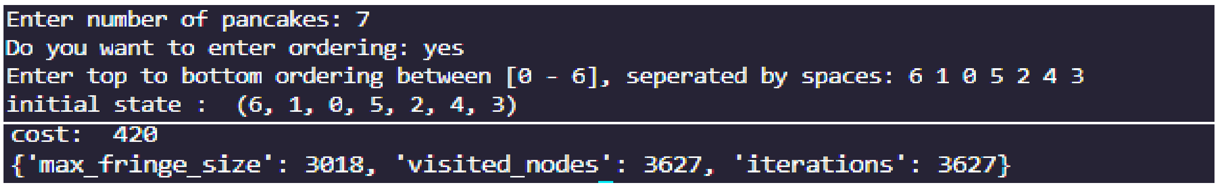


```
Enter number of pancakes: 8
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 7], seperated by spaces: 7 6 4 5 0 2 1 3
initial state :  (7, 6, 4, 5, 0, 2, 1, 3)
Iterative result : [(None, (7, 6, 4, 5, 0, 2, 1, 3)), (7, (3, 1, 2, 0, 5, 4, 6, 7)), (4, (5, 0, 2, 1, 3, 4, 6, 7)), (5, (4, 3, 1, 2, 0, 5, 6, 7)), (4, (0, 2,
1, 3, 4, 5, 6, 7)), (2, (1, 2, 0, 3, 4, 5, 6, 7)), (1, (2, 1, 0, 3, 4, 5, 6, 7)), (2, (0, 1, 2, 3, 4, 5, 6, 7))]
cost:  7
{'max_fringe_size': 42, 'visited_nodes': 26841, 'iterations': 26841}
```

## Breadth First:



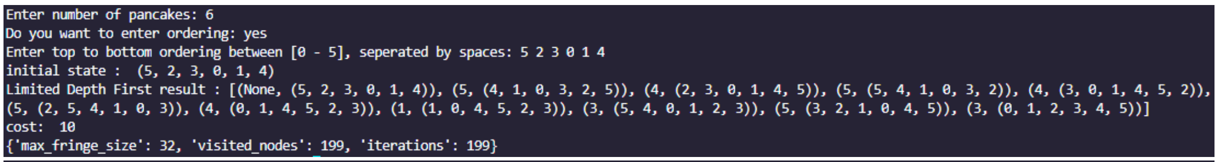
```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
cost:  154
{'max_fringe_size': 401, 'visited_nodes': 393, 'iterations': 393}
```



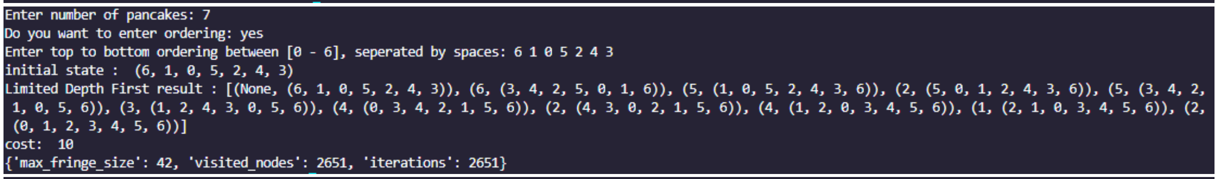
```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
cost:  420
{'max_fringe_size': 3018, 'visited_nodes': 3627, 'iterations': 3627}
```

*(It cannot reached to the result for 7 6 4 5 0 2 1 3 )*

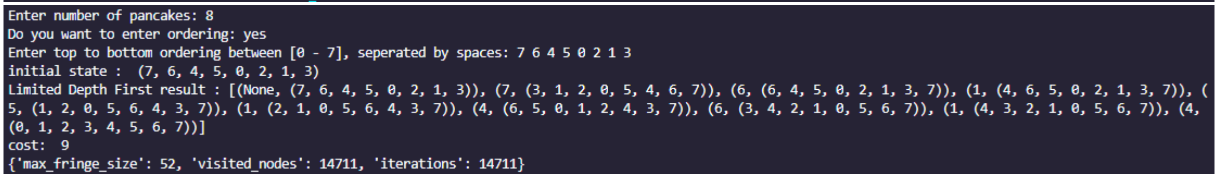
## Limited Depth First(dept_limit=10):



```
Enter number of pancakes: 6
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 5], seperated by spaces: 5 2 3 0 1 4
initial state :  (5, 2, 3, 0, 1, 4)
Limited Depth First result : [(None, (5, 2, 3, 0, 1, 4)), (5, (4, 1, 0, 3, 2, 5)), (4, (2, 3, 0, 1, 4, 5)), (5, (5, 4, 1, 0, 3, 2)), (4, (3, 0, 1, 4, 5, 2)),
(5, (2, 5, 4, 1, 0, 3)), (4, (0, 1, 4, 5, 2, 3)), (1, (1, 0, 4, 5, 2, 3)), (3, (5, 4, 0, 1, 2, 3)), (5, (3, 2, 1, 0, 4, 5)), (3, (0, 1, 2, 3, 4, 5))]
cost:  10
{'max_fringe_size': 32, 'visited_nodes': 199, 'iterations': 199}
```



```
Enter number of pancakes: 7
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 6], seperated by spaces: 6 1 0 5 2 4 3
initial state :  (6, 1, 0, 5, 2, 4, 3)
Limited Depth First result : [(None, (6, 1, 0, 5, 2, 4, 3)), (6, (3, 4, 2, 5, 0, 1, 6)), (5, (1, 0, 5, 2, 4, 3, 6)), (2, (5, 0, 1, 2, 4, 3, 6)), (5, (3, 4, 2,
 1, 0, 5, 6)), (3, (1, 2, 4, 3, 0, 5, 6)), (4, (0, 3, 4, 2, 1, 5, 6)), (2, (4, 3, 0, 2, 1, 5, 6)), (4, (1, 2, 0, 3, 4, 5, 6)), (1, (2, 1, 0, 3, 4, 5, 6)), (2,
 (0, 1, 2, 3, 4, 5, 6))]
cost:  10
{'max_fringe_size': 42, 'visited_nodes': 2651, 'iterations': 2651}
```



```
Enter number of pancakes: 8
Do you want to enter ordering: yes
Enter top to bottom ordering between [0 - 7], seperated by spaces: 7 6 4 5 0 2 1 3
initial state :  (7, 6, 4, 5, 0, 2, 1, 3)
Limited Depth First result : [(None, (7, 6, 4, 5, 0, 2, 1, 3)), (7, (3, 1, 2, 0, 5, 4, 6, 7)), (6, (6, 4, 5, 0, 2, 1, 3, 7)), (1, (4, 6, 5, 0, 2, 1, 3, 7)), (
5, (1, 2, 0, 5, 6, 4, 3, 7)), (1, (2, 1, 0, 5, 6, 4, 3, 7)), (4, (6, 5, 0, 1, 2, 4, 3, 7)), (6, (3, 4, 2, 1, 0, 5, 6, 7)), (1, (4, 3, 2, 1, 0, 5, 6, 7)), (4,
(0, 1, 2, 3, 4, 5, 6, 7))]
cost:  9
{'max_fringe_size': 52, 'visited_nodes': 14711, 'iterations': 14711}
```