

# **Term Project: Missionaries and Cannibals**

## **Problem**

Gaurav Kumar Jindal(gj3bg), Sanchit Sinha (ss7mu)

The missionaries and cannibals problem is a popular problem in Algorithms and Artificial Intelligence because it was discussed in the 1st paper that approached problem formulation from an analytical viewpoint (Amerel, 1968). In the problem, three missionaries and three cannibals are on the Left bank {L} of a river, along with a boat that can hold up to two people. We have to find a sequence which we should follow to transport missionaries and cannibals to Right bank {R} without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place (if they were, the cannibals will eat the missionaries). Also, we have to satisfy another constraint of having at least one person in the boat i.e.  $1 \leq \text{capacity of boat} \leq 2$  as the boat cannot cross the river by itself with no people on board.

We have generalised this problem to M: number of missionaries, C: number of cannibals, K: boat capacity. To solve this problem we have formulated this problem into a graph where state is represented by simple tuple  $\langle m, c, \text{dir} \rangle$  where m is the number of missionaries on the left bank {L}, c is the number of cannibals on the left bank {L}, dir is the direction of boat having value 1 for left to right and 0 for right to left. Our aim is to reach a state having value  $\langle 0, 0, 0 \rangle$  which means we have everyone on left bank crossed the river and boat is also at the right bank. We have solved this problem by using breadth first search (BFS) and depth first search (DFS).

The time complexity of BFS and DFS is the same that is  $O(V+E)$  where V and E are the vertex and edges of the graph. However, BFS always returns the shortest path.

```

sanchit@sanchit_mbp:~/Desktop/Masters/Algos/Project/AlgorithmProject$ python3 submission.py
Number of Missionaries: 3
Number of Cannibals: 3
Boat Capacity: 2
BFS

State: Left (M : 3, C : 3), Boat Position: (left), Right (M: 0, C: 0)

Move 1 M and 1 C from left to right.
State: Left (M : 2, C : 2), Boat Position: (right), Right (M: 1, C: 1)

Move 1 M and 0 C from right to left.
State: Left (M : 3, C : 2), Boat Position: (left), Right (M: 0, C: 1)

Move 0 M and 2 C from left to right.
State: Left (M : 3, C : 0), Boat Position: (right), Right (M: 0, C: 3)

Move 0 M and 1 C from right to left.
State: Left (M : 3, C : 1), Boat Position: (left), Right (M: 0, C: 2)

Move 2 M and 0 C from left to right.
State: Left (M : 1, C : 1), Boat Position: (right), Right (M: 2, C: 2)

Move 1 M and 1 C from right to left.
State: Left (M : 2, C : 2), Boat Position: (left), Right (M: 1, C: 1)

Move 2 M and 0 C from left to right.
State: Left (M : 0, C : 2), Boat Position: (right), Right (M: 3, C: 1)

Move 0 M and 1 C from right to left.
State: Left (M : 0, C : 3), Boat Position: (left), Right (M: 3, C: 0)

Move 0 M and 2 C from left to right.
State: Left (M : 0, C : 1), Boat Position: (right), Right (M: 3, C: 2)

Move 1 M and 0 C from right to left.
State: Left (M : 1, C : 1), Boat Position: (left), Right (M: 2, C: 2)

Move 0 M and 2 C from left to right.
State: Left (M : 0, C : 0), Boat Position: (right), Right (M: 3, C: 3)
Count = 11

```

```

DFS

State: Left (M : 3, C : 3), Boat Position: (left), Right (M: 0, C: 0)

Move 1 M and 1 C from left to right.
State: Left (M : 2, C : 2), Boat Position: (right), Right (M: 1, C: 1)

Move 1 M and 0 C from right to left.
State: Left (M : 3, C : 2), Boat Position: (left), Right (M: 0, C: 1)

Move 0 M and 2 C from left to right.
State: Left (M : 3, C : 0), Boat Position: (right), Right (M: 0, C: 3)

Move 0 M and 1 C from right to left.
State: Left (M : 3, C : 1), Boat Position: (left), Right (M: 0, C: 2)

Move 2 M and 0 C from left to right.
State: Left (M : 1, C : 1), Boat Position: (right), Right (M: 2, C: 2)

Move 1 M and 1 C from right to left.
State: Left (M : 2, C : 2), Boat Position: (left), Right (M: 1, C: 1)

Move 2 M and 0 C from left to right.
State: Left (M : 0, C : 2), Boat Position: (right), Right (M: 3, C: 1)

Move 0 M and 1 C from right to left.
State: Left (M : 0, C : 3), Boat Position: (left), Right (M: 3, C: 0)

Move 0 M and 2 C from left to right.
State: Left (M : 0, C : 1), Boat Position: (right), Right (M: 3, C: 2)

Move 1 M and 0 C from right to left.
State: Left (M : 1, C : 1), Boat Position: (left), Right (M: 2, C: 2)

Move 1 M and 1 C from left to right.
State: Left (M : 0, C : 0), Boat Position: (right), Right (M: 3, C: 3)
Count = 11

```

BFS and DFS outputs on Missionaries = 3 and Cannibals = 3, boat size = 2

```
[sanchit@sanchit_mbp:~/Desktop/Masters/Algos/Project/AlgorithmProject$ python3 submission.py
Number of Missionaries: 25
Number of Cannibals: 25
Boat Capacity: 23
BFS
```

```
State: Left (M : 25, C : 25), Boat Position: (left), Right (M: 0, C: 0)
Move 11 M and 11 C from left to right.
State: Left (M : 14, C : 14), Boat Position: (right), Right (M: 11, C: 11)
Move 11 M and 4 C from right to left.
State: Left (M : 25, C : 18), Boat Position: (left), Right (M: 0, C: 7)
Move 15 M and 8 C from left to right.
State: Left (M : 10, C : 10), Boat Position: (right), Right (M: 15, C: 15)
Move 1 M and 1 C from right to left.
State: Left (M : 11, C : 11), Boat Position: (left), Right (M: 14, C: 14)
Move 11 M and 11 C from left to right.
State: Left (M : 0, C : 0), Boat Position: (right), Right (M: 25, C: 25)
Count = 5
```

```
Move 11 M and 11 C from right to left.
State: Left (M : 14, C : 14), Boat Position: (left), Right (M: 11, C: 11)
Move 14 M and 9 C from left to right.
State: Left (M : 0, C : 5), Boat Position: (right), Right (M: 25, C: 20)
Move 13 M and 8 C from right to left.
State: Left (M : 13, C : 13), Boat Position: (left), Right (M: 12, C: 12)
Move 13 M and 10 C from left to right.
State: Left (M : 0, C : 3), Boat Position: (right), Right (M: 25, C: 22)
Move 12 M and 9 C from right to left.
State: Left (M : 12, C : 12), Boat Position: (left), Right (M: 13, C: 13)
Move 12 M and 11 C from left to right.
State: Left (M : 0, C : 1), Boat Position: (right), Right (M: 25, C: 24)
Move 11 M and 10 C from right to left.
State: Left (M : 11, C : 11), Boat Position: (left), Right (M: 14, C: 14)
Move 11 M and 11 C from left to right.
State: Left (M : 0, C : 0), Boat Position: (right), Right (M: 25, C: 25)
Count = 49
```

BFS and DFS outputs on Missionaries = 25 and Cannibals = 25, boat size = 23. As it can be seen that BFS finds the smallest number of moves.