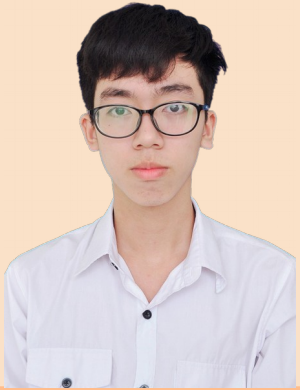
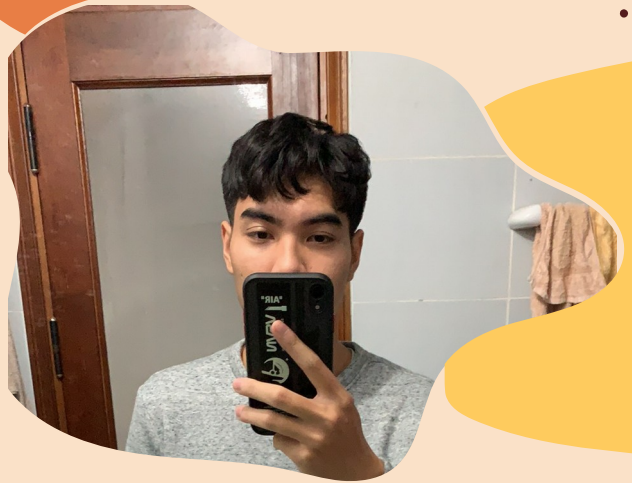
OUR TEAM



Le Duc Anh

20194416



Nguyen Huy Hoang

20194433

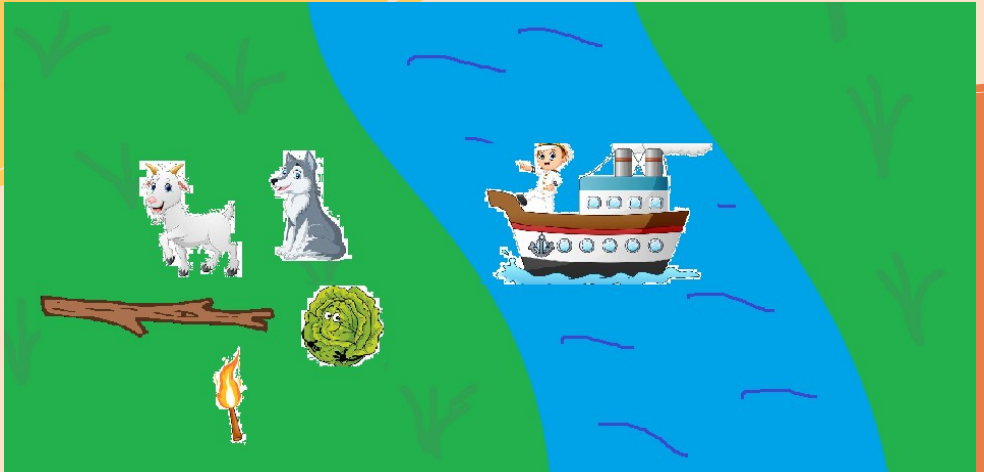
Extended wolf-goat-cabbage problem

Table of Contents

01. Problem description

- The shepherd must bring a wolf, a goat, a cabbage, a wooden stick and a fire torch on the other side of a river.
- Without the shepherd:
  - If the wolf and the goat are together, the wolf eats the goat.
  - If the goat and the cabbage are together, the goat eats the cabbage.
  - If the stick and the wolf are together, the stick beats the wolf.
  - If the stick and the torch are together, the torch burns the stick.
- The boat can transport 2 objects or animals across the river at each step.

Objective: Bring all of them to the other side



Problem formulation

- **States:** boat location + animals and objects location. There are 6 things (shepherd, goat, cabbage, wolf, wooden stick, fire torch) and each one has 2 status (on each side of the river).
- **Initial state:** The three animals, two objects and the boat are on the left side of the river.
- **Actions:** Bring up to 2 animals or objects at each time the boat crossing the river
- **Goal test**: All the animals and objects are on the other side of the river.
- **Path cost**: Each step cost 1, so path cost = number of steps in path.

02

Algorithms

Generating states

```
def gen_brand(visited, parent_map):
    if boat on the right:
        for i in left:
            new_left = left(copy)
            new_right = right(copy)
            remove i from left
            append i in right
            for j in left:
                next_left = new_left(copy)
                next right = new_right(copy)
                remove i from left
                append j in right
                if sorted(next_left) not conflict_states and not existed_states:
                    child = new_state
                    parent_map = current_state
            if sorted(new_left) not conflict_states and not existed_states:
                child = new_state
                parent_map = current_state
        if sorted(left) not conflict_states and not existed_states:
            child = new_state
            parent_map = current_state
    else:
        (quite similar)
```

Bringing things to the other side by removing it from the array left and appending it to to array right

Check the state when bringing 2 things

Check the state when bringing 1 things

Check the state when the shepherd go alone

Do the same when the boat is on the left

Breadth-first Search
Depth-first Search

BFS	DFS
- Check all the states - Give optimal solution	- Much faster - Return solution but not optimal

03 Result analysis

BFS Solution

[W, G, C, S, T]~~~[]- The boat is on the Left

[G, C, S]~~~[T, W]- The boat is on the Right

[G, C, S]~~~[T, W]- The boat is on the Left

[G]~~~[T, W, S, C]- The boat is on the Right

[G, S]~~~[T, W, C]- The boat is on the Left

[]~~~[T, W, C, S, G]- The boat is on the Right

NOTATION

W: the wolf
G: the goat
C: the cabbage
S: the wooden stick
T: the fire torch

DFS Solution

[W, G, C, S, T]~~~[]- The boat is on the Left

[G, C, S]~~~[T, W]- The boat is on the Right

[G, C, S]~~~[T, W]- The boat is on the Left

[G, C]~~~[T, W, S]- The boat is on the Right

[G, C, S, W]~~~[T]- The boat is on the Left

[G, S]~~~[T, W, C]- The boat is on the Right

[G, S]~~~[T, W, C]- The boat is on the Left

[G]~~~[T, W, C, S]- The boat is on the Right

[G, S, W]~~~[T, C]- The boat is on the Left

[S]~~~[T, C, W, G]- The boat is on the Right

[S, G, T]~~~[C, W]- The boat is on the Left

[T]~~~[C, W, G, S]- The boat is on the Right

[T, S, W]~~~[C, G]- The boat is on the Left

[‘T, W]~~~[C, G, S]- The boat is on the Right

[T, W]~~~[C, G, S]- The boat is on the Left

[]~~~[C, G, S, W, T]- The boat is on the Right

NOTATION

W: the wolf
G: the goat
C: the cabbage
S: the wooden stick
T: the fire torch

	Complete ?	Time	Space	Optimal?
DFS	Yes	600	711	Yes
BFS	Yes	16	88	No

Breadth-first Search

The running time and the space complexity of BFS are much larger but the solution is optimal.

Depth-first Search

The running time and the space complexity of BFS are much smaller but the solution is not optimal.

WHOA!

FINAL CONCLUSION!

- Formulating a problem solved by searching
- Analyzing and implementing search algorithms
- Team work
- Researching skills
- Writing report skills

THANKS

Does anyone have any questions?

huyhoang240101@gmail.com
0949681098

Q & A?