

## ĐỀ BÀI:

Cho 3 bình nước (bình 1: 8 lít, bình 2: 5 lít, bình 3: 3 lít), yêu cầu đo 4 lít.

Hãy giải bài toán trên bằng lập trình logic.

## BÀI LÀM

### 1. Số lượng đối tượng tham gia vào trạng thái của bài toán:

- Gồm 3 bình nước, với bình 1 có dung tích tối đa là 8 lít, bình 2 có dung tích tối đa là 5 lít, bình 3 có dung tích tối đa là 3 lít

### 2. Biểu diễn trạng thái của bài toán:

Định nghĩa một trạng thái:  $\text{State}(B1, B2, B3)$ .

Một trạng thái được biểu diễn dưới dạng danh sách là  $(B1, B2, B3)$  với mỗi phần tử được định nghĩa là:

- B1: khối lượng nước đang có trong bình 1.
- B2: khối lượng nước đang có trong bình 2.
- B3: khối lượng nước đang có trong bình 3.

Ví dụ,  $\text{State}(8, 5, 3)$  có nghĩa là:

- 8: khối lượng nước đang có trong bình 1 là 8 lít.
- 5: khối lượng nước đang có trong bình 2 là 5 lít.
- 3: khối lượng nước đang có trong bình 3 là 3 lít.

Trạng thái ban đầu:  $\text{initial\_state}(0, 0, 0)$  - tất cả các bình đều ở trạng thái rỗng.

Trạng thái đích:

$\text{goal\_state}(4, \_, \_)$  - Bình 1 có dung tích đạt được là 4 lít, các bình còn lại chứa bao nhiêu nước cũng được.

$\text{goal\_state}(\_, 4, \_)$  - Bình 2 có dung tích đạt được là 4 lít, các bình còn lại chứa bao nhiêu nước cũng được.

### 3. Các thao tác / hành động có thể tác động lên các trạng thái:

Các phép toán mô tả các thao tác gồm:

- Đổ đầy bình 1.
- Đổ đầy bình 2.
- Đổ đầy bình 3.

4. Làm rỗng bình 1.
5. Làm rỗng bình 2.
6. Làm rỗng bình 3.
7. Đổ từ bình 1 sang bình 2.
8. Đổ từ bình 1 sang bình 3.
9. Đổ từ bình 2 sang bình 1.
10. Đổ từ bình 2 sang bình 3.
11. Đổ từ bình 3 sang bình 1.
12. Đổ từ bình 3 sang bình 2.

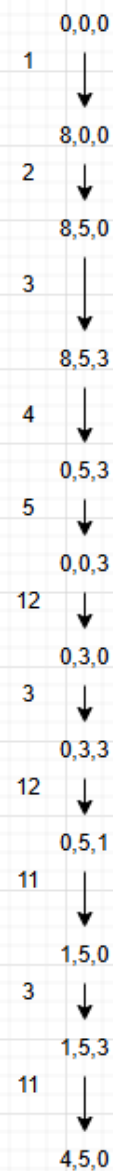
#### **4. Mô tả không gian trạng thái:**

Ràng buộc:

- Số lượng nước trong mỗi bình không được nhỏ hơn 0 và không được nhiều hơn khối lượng nước tối đa mà bình đó có thể chứa được. Bình 1 khối lượng nước tối đa có thể chứa là 8 lít, bình 2 là 5 lít, bình 3 là 3 lít.
- Mỗi phép toán trên có thể thực hiện nếu trạng thái tiếp theo là an toàn và chưa được thực hiện trước đó.

Hình ảnh:

- 1. Đổ đầy bình 1.
  2. Đổ đầy bình 2.
  3. Đổ đầy bình 3.
  4. Làm rỗng bình 1.
  5. Làm rỗng bình 2.
  6. Làm rỗng bình 3.
  7. Đổ nước từ bình 1 sang bình 2.
  8. Đổ nước từ bình 1 sang bình 3.
  9. Đổ nước từ bình 2 sang bình 1.
  10. Đổ nước từ bình 2 sang bình 3.
  11. Đổ nước từ bình 3 sang bình 1.
  12. Đổ nước từ bình 3 sang bình 2.



Không gian trạng thái:

Gồm tất cả các cách phân phối lượng nước trong ba bình với dung tích tối đa lần lượt là 8 lít, 5 lít và 3 lít. Vì vậy, các giá trị khả thi cho mỗi biến là:

- B1: từ 0 đến 8 (dung tích của bình 1),
- B2: từ 0 đến 5 (dung tích của bình 2),
- B3: từ 0 đến 3 (dung tích của bình 3).

Vì thế, tổng số trạng thái trong không gian là  $9 \times 6 \times 4 = 216$  trạng thái

## 5. Áp dụng thuật toán DFS để tìm nghiệm:

**Bước 1:**

**OPEN:**  $\{(0, 0, 0)\}$

**CLOSED:**  $\{\}$

**Bước 2:**

Trạng thái đang xét:  $(0, 0, 0)$

Trạng thái mới sinh ra:  $(8, 0, 0)$

**OPEN:**  $\{(8, 0, 0)\}$

**CLOSED:**  $\{(0, 0, 0)\}$

**Bước 3:**

Trạng thái đang xét:  $(8, 0, 0)$

Trạng thái mới sinh ra:  $(8, 5, 0)$

**OPEN:**  $\{(8, 5, 0)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0)\}$

**Bước 4:**

Trạng thái đang xét:  $(8, 5, 0)$

Trạng thái mới sinh ra:  $(8, 5, 3)$

**OPEN:**  $\{(8, 5, 3)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0)\}$

**Bước 5:**

Trạng thái đang xét:  $(8, 5, 3)$

Trạng thái mới sinh ra:  $(0, 5, 3)$

**OPEN:**  $\{(0, 5, 3)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3)\}$

**Bước 6:**

Trạng thái đang xét:  $(0, 5, 3)$

Trạng thái mới sinh ra:  $(0, 0, 3)$

**OPEN:**  $\{(0, 0, 3)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3)\}$

**Bước 7:**

Trạng thái đang xét:  $(0, 0, 3)$

Trạng thái mới sinh ra:  $(3, 0, 0)$

**OPEN:**  $\{(3, 0, 3)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3), (0, 0, 3)\}$

**Bước 8:**

Trạng thái đang xét:  $(3, 0, 3)$

Trạng thái mới sinh ra:  $(3, 3, 0)$

**OPEN:**  $\{(3, 3, 0)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3), (0, 0, 3), (3, 0, 3)\}$

**Bước 9:**

Trạng thái đang xét:  $(3, 3, 0)$

Trạng thái mới sinh ra:  $(6, 0, 3)$

**OPEN:**  $\{(6, 0, 3)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3), (0, 0, 3), (3, 0, 3), (3, 3, 0)\}$

**Bước 10:**

Trạng thái đang xét:  $(6, 0, 3)$

Trạng thái mới sinh ra:  $(6, 3, 0)$

**OPEN:**  $\{(6, 3, 0)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3), (0, 0, 3), (3, 0, 3), (3, 3, 0), (6, 0, 3)\}$

**Bước 11:**

Trạng thái đang xét:  $(6, 3, 0)$

Trạng thái mới sinh ra:  $(8, 1, 0)$

**OPEN:**  $\{(8, 1, 0)\}$

**CLOSED:**  $\{(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3), (0, 0, 3), (3, 0, 3), (3, 3, 0), (6, 0, 3), (6, 3, 0)\}$

### Bước 12:

Trạng thái đang xét: (8, 1, 0)

Trạng thái mới sinh ra: (4, 5, 0)

**OPEN:** {(4, 5, 0)}

**CLOSED:** {(0, 0, 0), (8, 0, 0), (8, 5, 0), (8, 5, 3), (0, 5, 3), (0, 0, 3), (3, 0, 3), (3, 3, 0), (6, 0, 3), (6, 3, 0), (8, 1, 0)}

### Bước 13:

Trạng thái đang xét: (4, 5, 0)

Đây là trạng thái đích vì bình 1 có 4 lít nước.

=> **Kết thúc tìm kiếm.**

## 6. Chương trình Prolog:

```
initial_state((0, 0, 0)).
```

```
goal_state((4, _, _)).
```

```
goal_state(_, 4, _).
```

```
pour_full_1((X, Y, Z), (8, Y, Z)) :- X < 8.
```

```
pour_full_2((X, Y, Z), (X, 5, Z)) :- Y < 5.
```

```
pour_full_3((X, Y, Z), (X, Y, 3)) :- Z < 3.
```

```
pour_empty_1((X, Y, Z), (0, Y, Z)) :- X > 0.
```

```
pour_empty_2((X, Y, Z), (X, 0, Z)) :- Y > 0.
```

```
pour_empty_3((X, Y, Z), (X, Y, 0)) :- Z > 0.
```

```
pour_1_to_2((X, Y, Z), (X1, Y1, Z)) :-
```

```
    X > 0,
```

```
    Y < 5,
```

```
    Transfer is min(X, 5 - Y),
```

```
    X1 is X - Transfer,
```

```
    Y1 is Y + Transfer.
```

`pour_1_to_3((X, Y, Z), (X1, Y, Z1)) :-`

`X > 0,`  
`Z < 3,`  
`Transfer is min(X, 3 - Z),`  
`X1 is X - Transfer,`  
`Z1 is Z + Transfer.`

`pour_2_to_1((X, Y, Z), (X1, Y1, Z)) :-`

`Y > 0,`  
`X < 8,`  
`Transfer is min(Y, 8 - X),`  
`X1 is X + Transfer,`  
`Y1 is Y - Transfer.`

`pour_2_to_3((X, Y, Z), (X, Y1, Z1)) :-`

`Y > 0,`  
`Z < 3,`  
`Transfer is min(Y, 3 - Z),`  
`Y1 is Y - Transfer,`  
`Z1 is Z + Transfer.`

`pour_3_to_1((X, Y, Z), (X1, Y, Z1)) :-`

`Z > 0,`  
`X < 8,`  
`Transfer is min(Z, 8 - X),`  
`X1 is X + Transfer,`  
`Z1 is Z - Transfer.`

`pour_3_to_2((X, Y, Z), (X, Y1, Z1)) :-`

`Z > 0,`  
`Y < 5,`  
`Transfer is min(Z, 5 - Y),`  
`Y1 is Y + Transfer,`  
`Z1 is Z - Transfer.`

`adjacent_states((X, Y, Z), AdjList) :-`

`findall((X1, Y1, Z1),`  
`(`  
`pour_full_1((X, Y, Z), (X1, Y1, Z1));`

```

    pour_full_2((X, Y, Z), (X1, Y1, Z1));
    pour_full_3((X, Y, Z), (X1, Y1, Z1));
    pour_empty_1((X, Y, Z), (X1, Y1, Z1));
    pour_empty_2((X, Y, Z), (X1, Y1, Z1));
    pour_empty_3((X, Y, Z), (X1, Y1, Z1));
    pour_1_to_2((X, Y, Z), (X1, Y1, Z1));
    pour_1_to_3((X, Y, Z), (X1, Y1, Z1));
    pour_2_to_1((X, Y, Z), (X1, Y1, Z1));
    pour_2_to_3((X, Y, Z), (X1, Y1, Z1));
    pour_3_to_1((X, Y, Z), (X1, Y1, Z1));
    pour_3_to_2((X, Y, Z), (X1, Y1, Z1))
),
AdjList).

```

```

dfs((X, Y, Z), Visited, [(X, Y, Z)|Visited]) :-
    goal_state((X, Y, Z)), !.

```

```

dfs((X, Y, Z), Visited, Path) :-
    adjacent_states((X, Y, Z), AdjList),
    member((X1, Y1, Z1), AdjList),
    \+ member((X1, Y1, Z1), Visited),
    dfs((X1, Y1, Z1), [(X, Y, Z)|Visited], Path).

```

```

solve_puzzle(Path) :-
    initial_state(InitialState),
    dfs(InitialState, [], Path).

```