

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



---

**BÀI TẬP MÔN TRÍ TUỆ NHÂN TẠO**

---

**BÀI TẬP 1 – DFS/BFS/UCS for Sokoban**

**GV hướng dẫn:**

TS. Lương Ngọc Hoàng

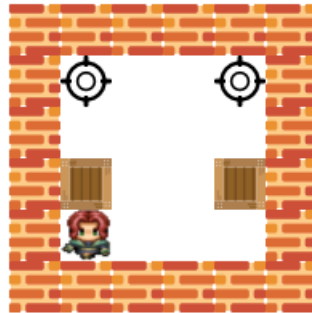
**Sinh viên thực hiện:**

Trương Phúc Trường – 22521587

TP. HCM, ngày 8 tháng 3 năm 2024

---

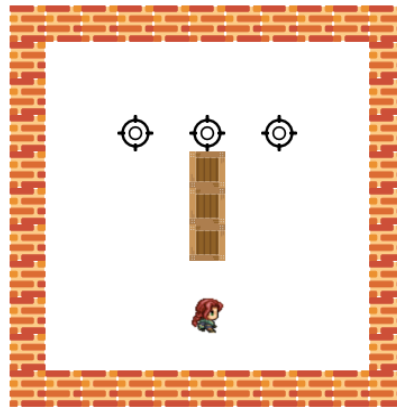
# 1. Mô tả trò chơi và nguyên tắc



- Sokoban là trò chơi mà người chơi điều khiển một nhân vật di chuyển trên bản đồ được chia thành các ô vuông và phải đẩy các thùng hàng về ô vuông đích (như hình trên là các hình tròn lồng nhau).
- Trạng thái khởi đầu là không gian trò chơi có nhân vật, thùng hàng và ô vuông đích được quy định theo màn chơi lúc vừa bắt đầu.
- Trạng thái kết thúc là khi các thùng hàng đã được đặt tại các ô vuông đích.
- Nguyên tắc: mỗi thùng hàng có thể được đặt ở bất kì chỗ nào, chúng được đẩy bởi nhân vật và nhân vật sẽ di chuyển theo 4 hướng trái, phải, lên, xuống (l, r, u, d), khi đẩy thùng hàng cũng sẽ là 4 hướng đó nhưng được kí hiệu khác (L, R, U, D).
- Lưu ý: các thùng hàng không thể đẩy xuyên tường cũng như không thể đẩy nhiều thùng cùng lúc.
- Có những trường hợp trò chơi không thể giải được do đẩy thùng hàng vào góc tường hoặc đẩy vào những vị trí mà nhân vật chỉ có thể đẩy được thùng hàng theo 2 hướng.

## 2. Mô hình hóa

```
#####↓
#           #↓
#           #↓
#   .   .   #↓
#   B     #↓
#   B     #↓
#   B     #↓
#   &     #↓
#           #↓
#####↓
```



Ví dụ mô phỏng level 5 của trò chơi

### 2.1 Mô phỏng trò chơi:

Sơ đồ bài toán sẽ được lưu trữ dưới dạng một file \*.txt với:

- + '#': Tường chắn
- + 'B': thùng hàng
- + '&': nhân vật
- + '.': vị trí đích

### 2.2 Không gian trạng thái:

– Trong trò Sokoban, **không gian trạng thái** là một tập hợp trừu tượng mô tả tất cả các cấu hình có thể xảy ra của bản đồ trò chơi. Mỗi cấu hình đại diện cho vị trí của nhân vật và các thùng hàng trên bản đồ.

– Thành phần:

+ *Trạng thái*: Mỗi trạng thái trong không gian trạng thái được biểu diễn bởi một tuple bao gồm:

- Vị trí của nhân vật (x, y)
- Vị trí của từng thùng hàng (ví dụ: [(x1, y1), (x2, y2),...])

+ *Phép toán*: Các phép toán xác định cách chuyển đổi giữa các trạng thái:

- Di chuyển nhân vật lên, xuống, trái, phải

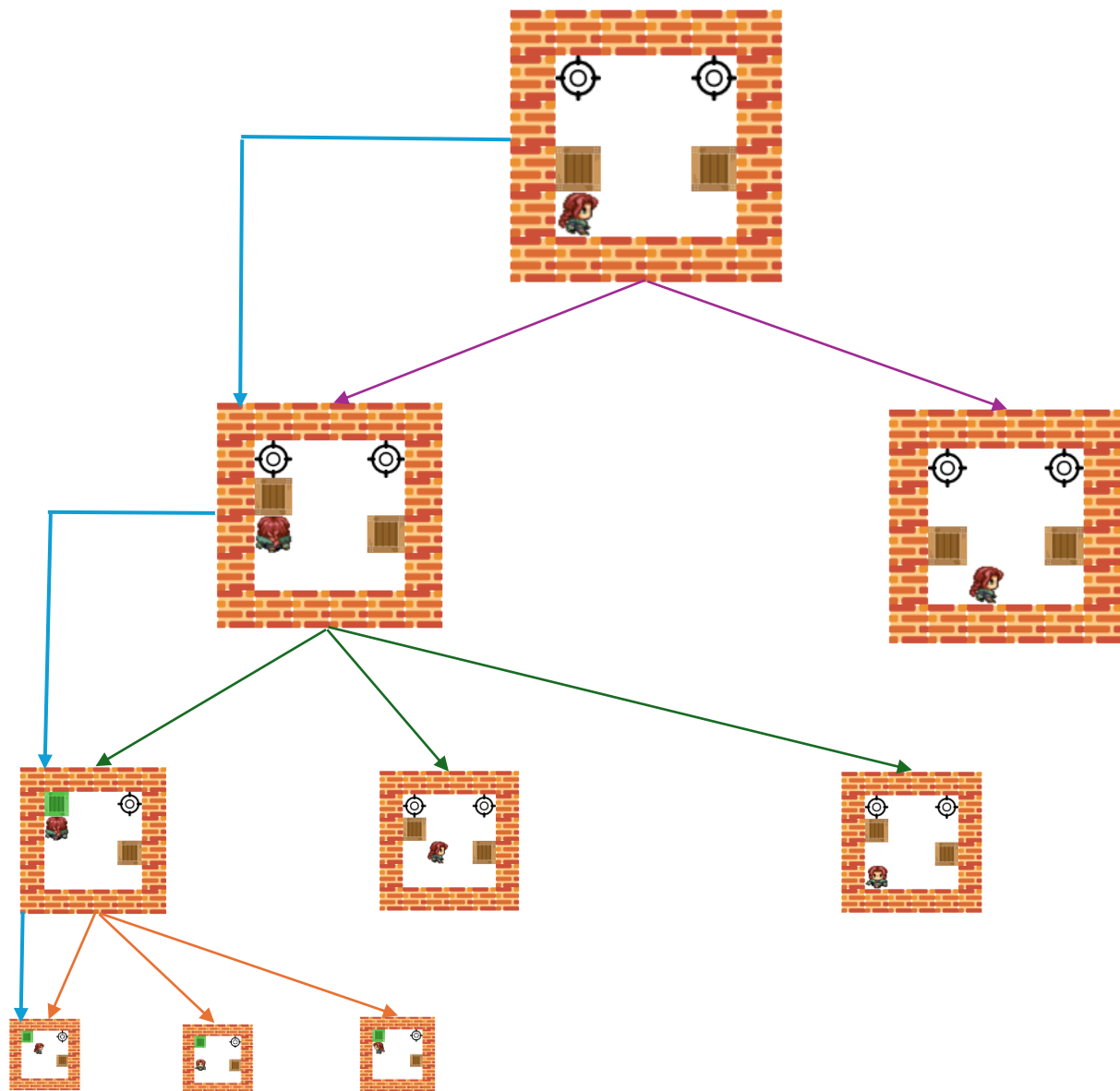
- Đẩy thùng hàng theo hướng di chuyển của nhân vật

### 3. Hàm tiến triển

Chúng ta sẽ áp dụng 3 thuật toán tìm kiếm sau đây để giải trò sokoban

#### 3.1 Depth First Search (Tìm kiếm theo chiều sâu)

- Thuật toán DFS bắt đầu từ trạng thái ban đầu và khám phá các nhánh con theo chiều sâu.
- Nó lưu trữ một ngăn xếp để theo dõi các trạng thái đã truy cập.
- Khi một trạng thái được truy cập, nó sẽ được thêm vào stack.
- Thuật toán sẽ tiếp tục khám phá các trạng thái con cho đến khi tìm thấy trạng thái mục tiêu hoặc stack rỗng.
- Nếu stack rỗng, nghĩa là không có giải pháp cho trò chơi.
- *Ưu điểm:*
  - + Đơn giản để triển khai.
  - + Có thể tìm ra giải pháp nhanh nếu có giải pháp gần trạng thái ban đầu.
- *Nhược điểm:*
  - + Có thể tốn nhiều bộ nhớ nếu có nhiều nhánh con.
  - + Có thể không tìm ra giải pháp tối ưu.

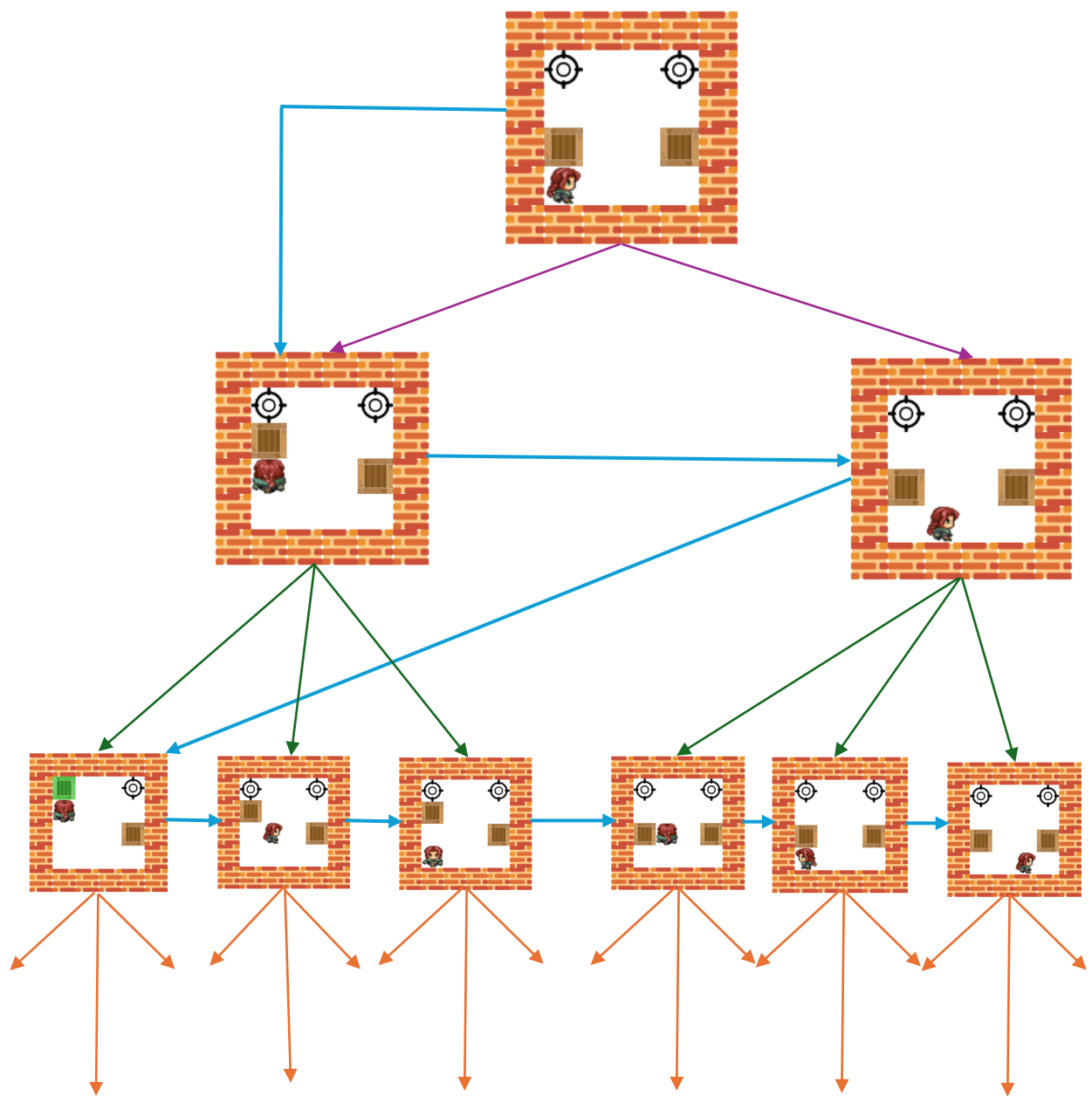


Ví dụ mô tả đồ thị trạng thái trò chơi level 2. Nút gốc là trạng thái bắt đầu. Theo mũi tên tím, xanh lá và cam lần lượt biểu diễn quan hệ các nút trạng thái. Mũi tên xanh dương là mẫu trình tự duyệt theo thuật toán DFS

### 3.2 Breadth First Search (Tìm kiếm theo chiều rộng)

- Thuật toán BFS bắt đầu từ trạng thái ban đầu và khám phá tất cả các trạng thái cùng độ sâu trước khi chuyển sang độ sâu tiếp theo.
- Nó sử dụng một hàng đợi để lưu trữ các trạng thái cần khám phá.
- Khi một trạng thái được truy cập, nó sẽ được thêm vào queue.
- Thuật toán sẽ tiếp tục khám phá các trạng thái con cho đến khi tìm thấy trạng thái mục tiêu hoặc queue rỗng.
- Nếu queue rỗng, nghĩa là không có giải pháp cho trò chơi.
- *Ưu điểm:*
  - + Luôn tìm ra giải pháp tối ưu (số bước ít nhất).
  - + Ít tốn bộ nhớ hơn DFS.
- *Nhược điểm:*

Có thể tốn nhiều thời gian hơn DFS nếu có nhiều trạng thái cùng độ sâu.



Ví dụ mô tả đồ thị trạng thái trò chơi level 2. Nút gốc là trạng thái bắt đầu. Theo mũi tím, xanh lá và cam lần lượt biểu diễn quan hệ các nút trạng thái. Mũi tên xanh dương là mẫu trình tự duyệt theo thuật toán BFS

### 3.3 UCS – Uniform Cost Search (Tìm kiếm chi phí đồng nhất)

- Thuật toán UCS sử dụng **hàm chi phí** để đánh giá độ "ước tính" của mỗi trạng thái đến trạng thái mục tiêu.
- Nó sử dụng một **priority queue** (hàng đợi ưu tiên) để lưu trữ các trạng thái cần khám phá, sắp xếp theo thứ tự chi phí ước tính.
- Khi một trạng thái được truy cập, nó sẽ được thêm vào priority queue.
- Thuật toán sẽ tiếp tục khám phá các trạng thái con cho đến khi tìm thấy trạng thái mục tiêu hoặc priority queue rỗng.
- Nếu priority queue rỗng, nghĩa là không có giải pháp cho trò chơi.
- *Ưu điểm:*
  - + Luôn tìm ra giải pháp tối ưu (chi phí ít nhất).
  - + Hiệu quả hơn BFS trong nhiều trường hợp.
- *Nhược điểm:*

Có thể tốn nhiều bộ nhớ hơn BFS nếu có nhiều trạng thái có cùng chi phí ước tính.



#### 4. Bảng thống kê về độ dài đường đi (số bước đi) tìm được bởi 3 thuật toán DFS, BFS và UCS tại tất cả các bản đồ có sẵn

Thuật toán Level	DFS	BFS	UCS
1	79	12	12
2	24	9	9
3	403	15	15
4	27	7	7
5	KGD	20	20
6	55	19	19
7	707	21	21
8	323	97	97
9	74	8	8
10	37	33	33
11	36	34	34
12	109	23	23
13	196	31	31
14	865	23	23
15	291	105	105
16	KGD	34	34
17	KGD	KGD	KGD
18	KGD	KGD	KGD

\*KGD: không giải được do cấu hình máy và cũng như thuật toán mất nhiều thời gian để giải và cũng có trường hợp không giải được.

– *Nhận xét:*

+ Theo em thấy thuật toán DFS có phần nhỉnh hơn về mặt tốc độ đối với những màn chơi dễ (đưa ra lời giải nhanh), nhưng lại tốn quá nhiều bước đi để đạt được trạng thái đích.

+ BFS và UCS cho ra kết quả rất giống nhau về số bước đi.

⇒ Nhìn chung em thấy thuật toán UCS vượt trội nhất trong 3 thuật toán này vì có thể là số bước đi giải ra so với BFS là không chênh lệch nhưng so hiệu quả về mặt thời gian thì UCS vẫn nhỉnh hơn.

– Trong các bản đồ, em nghĩ bản đồ 18 là khó giải nhất vì sau một khoảng thời gian chờ rất lâu nhưng không một thuật toán nào cho ra kết quả còn bản đồ màn 17 thì cả 3 thuật toán đều trả về kết quả “[ ]” trong khoảng thời gian khá ngắn nhưng với sự phỏng đoán của mình, em vẫn nghĩ bản đồ màn 18 là khó nhất.