**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN – ĐHQG TPHCM**

**KHOA: KHOA HỌC MÁY TÍNH**

🙞🕮🙜

**Ảnh có chứa Đồ họa, biểu tượng, hình mẫu, thiết kế đồ họa

Mô tả được tạo tự động**

**BÀI BÁO CÁO**

**BÀI TẬP 1**

***NỘI DUNG: DFS, BFS, UCS FOR SOKOBAN***

LỚP HỌC PHẦN: **CS106.O22**

GIẢNG VIÊN HƯỚNG DẪN: **LƯƠNG NGỌC HOÀNG**

SINH VIÊN THỰC HÀNH: **HUỲNH DANH ĐẠT**

MÃ SỐ SINH VIÊN: **22520211**

# Sokoban là gì ?

* **Sokoban** là trò chơi dạng câu đố trong đó người chơi phải đẩy một số khối vuông vượt qua chướng ngại vật để đến đích. Trò chơi được thiết kế vào năm 1981 bởi Hiroyuki Imabayashi và được ra mắt lần đầu vào tháng 12 năm 1982 (Wikipedia)

# Mô hình hoá sokoban:

* **Trạng thái khởi đầu:** Trạng thái bắt đầu của trò chơi với bản đồ cho trước có vị trí của các hộp và nhân vật được đặt ở vị trí khác nhau. Mục tiêu của người chơi là di chuyển các hộp đến các vị trí chỉ định
* **Trạng thái kết thúc:** Trạng thái kết thúc xảy ra khi tất cả các hộp được di chuyển tới vị trí chỉ định và trò chơi sẽ kết thúc.
* **Không gian trạng thái:** Không gian trạng thái của Sokoban bao gồm tất cả các trạng thái vị trí của người chơi và các hộp có thể xảy ra. Mỗi trạng thái xác định bằng vị trí của người chơi và các hộp trên bản đồ
* **Các hành động hợp lệ:** Bao gồm các hành động như di chuyển nhân vật đẩy hoặc không đẩy thùng. Các bước đi bao gồm lên xuống trái phải, nhưng có một số trường hợp không hợp lệ như đi xuyên chướng ngại vật hay di chuyển cùng lúc hai hộp
* **Hàm tiến triển (successorr function):** Hàm này dùng để định nghĩa các trạng thái mới có thể xảy ra từ một trạng thái hiện tại sau khi thực hiện các hành động hợp lệ. Trong sokoban, hàm tiến triển sẽ trả về một tập hợp tạng thái mới có thể đạt được bằng cá hành động hợp lệ

|  |  |  |  |
| --- | --- | --- | --- |
| Levels | DFS | BFS | UCS |
| 1 | 79 (0.05s) | 12 (0.07s) | 12 (0.04s) |
| 2 | 24 (0.0s) | 9 (0.0s) | 9 (0.0s) |
| 3 | 403 (0.22s) | 15 (0.14s) | 15 (0.07s) |
| 4 | 27 (0.0s) | 7 (0.0s) | 7 (0.0s) |
| 5 | Inf | 20 (177.5s) | 20 (47.25s) |
| 6 | 55 (0.01s) | 19 (0.01s) | 19 (0.01s) |
| 7 | 707 (0.47s) | 21 (0.79s) | 21 (0.47s) |
| 8 | 323 (0.06s) | 97 (0.17s) | 97 (0.26s) |
| 9 | 74 (0.26s) | 8 (0.07s) | 8 (0.01s) |
| 10 | 37 (0.01s) | 33 (0.01s) | 33 (0.01s) |
| 11 | 36 (0.01s) | 34 (0.01s) | 34 (0.01s) |
| 12 | 109 (0.11s) | 23 (0.09s) | 23 (0.07s) |
| 13 | 185 (0.15s) | 31 (0.12s) | 31 (0.14s) |
| 14 | 865 (3.96s) | 23 (2.52s) | 23 (2.83s) |
| 15 | 291 (0.14s) | 105 (0.28s) | 105 (0.28s) |
| 16 | Inf | 34 (22.27s) | 34 (13.27s) |
| 17 | Inf | Inf | Inf |
| 18 | Inf | Inf | Inf |

(Các màn có chữ Inf khi chạy bị đứng máy)

# Nhận xét:

* **Thời gian:** Thuật toán UCS có thời gian chạy tối ưu nhất trong số cả 3 thuật toán, và thuật toán DFS có thời gian chạy lâu nhất
* **Số bước đi:** Thuật toán DFS có số bước đi lớn nhất trong khi đó cả hai thuật toán còn lại có số bước đi là như nhau
* **Thuật toán UCS tối ưu nhất khi cho ra số bước đi cũng như thời gian chạy ngắn nhất trong cả 3 thuật toán**
* Trong đó màn chơi có độ khó cao nhất là màn 17 và 18 khi cả đều có bản đồ rộng dẫn đến không gian trạng thái rất lớn và tốn rất nhiều thời gian cũng như chi phí và với cấu hình hiện tại của máy em không thể chạy được các màn đó