

# BÀI TẬP LÝ THUYẾT TUẦN 3

## Nhập môn Trí tuệ Nhân tạo

Nguyễn Hồng Yến – MSSV: 23280099

GVHD: PGS.TS Nguyễn Thanh Bình

### Phân tích các giải thuật tìm kiếm: BFS, DFS, UCS

#### Thuộc tính của DFS

<b>Complete?</b>	<b>Không.</b> DFS có thể đi vào nhánh vô hạn và không quay lại; chỉ hoàn thành nếu không gian trạng thái hữu hạn và có loại bỏ trạng thái lặp.
<b>Time</b>	$O(b^m)$ , với $b$ là hệ số phân nhánh và $m$ là độ sâu tối đa của không gian trạng thái.
<b>Space</b>	$O(bm)$ , do chỉ lưu đường đi hiện tại và tối đa $b$ nút tại mỗi mức sâu.
<b>Optimal?</b>	<b>Không.</b> DFS có thể tìm thấy lời giải sâu hơn dù tồn tại lời giải nông hơn; không đảm bảo chi phí nhỏ nhất ngay cả khi chi phí bước đi bằng nhau.

#### Lý do ngắn gọn

- Complete: nhánh vô hạn làm tìm kiếm không bao giờ quay lại các nhánh khác để gặp mục tiêu.
- Time: trong trường hợp xấu nhất mở rộng tất cả nút đến độ sâu  $m$ .
- Space: ngăn xếp tìm kiếm lưu tối đa  $m$  mức, mỗi mức tối đa  $b$  nút biên.
- Optimal: mở rộng theo chiều sâu, không theo chi phí hay độ sâu nhỏ nhất.

## Thuộc tính của BFS

<b>Complete?</b>	<b>Có.</b> BFS luôn tìm được lời giải nếu không gian trạng thái hữu hạn, vì nó duyệt theo từng mức độ sâu tăng dần.
<b>Time</b>	$O(b^{d+1})$ , trong đó $b$ là hệ số phân nhánh và $d$ là độ sâu của lời giải nông nhất.
<b>Space</b>	$O(b^{d+1})$ , do cần lưu toàn bộ các nút trong hàng đợi (fringe) của các mức đã mở rộng.
<b>Optimal?</b>	<b>Có</b> , nếu chi phí của mỗi bước đi là như nhau, BFS sẽ tìm ra lời giải có độ sâu nhỏ nhất.

## Lý do ngắn gọn

- Complete: Duyệt theo độ sâu tăng dần nên không bỏ sót lời giải nào.
- Time: Số lượng nút mở rộng tăng theo lũy thừa bậc  $d$ .
- Space: Cần lưu toàn bộ nút của các mức trước đó trong hàng đợi FIFO.
- Optimal: Lời giải đầu tiên được tìm thấy là lời giải nông nhất khi chi phí đồng nhất.

## Thuộc tính của UCS

<b>Complete?</b>	<b>Có.</b> UCS luôn tìm được lời giải nếu chi phí của mỗi bước đi dương (lớn hơn 0).
<b>Time</b>	$O(b^{1+\lceil C^*/\epsilon \rceil})$ , với $C^*$ là chi phí của lời giải tối ưu và $\epsilon$ là chi phí bước đi nhỏ nhất.
<b>Space</b>	$O(b^{1+\lceil C^*/\epsilon \rceil})$ , do cần lưu tất cả các nút trong hàng đợi ưu tiên (priority queue).
<b>Optimal?</b>	<b>Có.</b> UCS đảm bảo tìm được đường đi có chi phí thấp nhất (tối ưu toàn cục).

## Lý do ngắn gọn

- Complete: Không bị kẹt ở nhánh vô hạn nếu mọi chi phí bước đều dương.
- Time: Phụ thuộc vào số lượng nút có chi phí nhỏ hơn chi phí tối ưu.

- Space: Cần lưu trữ tất cả nút sinh ra trước khi mở rộng nút có chi phí thấp nhất.
- Optimal: Mở rộng nút theo chi phí tăng dần nên luôn tìm được đường đi chi phí nhỏ nhất.