

## تمرین سری اول درس هوش مصنوعی

۹۶۱۳۰۲۷

علی علی محمدی

### سؤال ۱:

- (آ) **نادرست**: زیرا عامل نمی‌داند که کلمه‌ی بعدی چیست یا با چه لحن و آهنگی بیان می‌شود یا در یک جمله‌ی خاص به چه معناست. این به دلیل خاصیت غیرقابل پیش‌بینی بودن زبان طبیعی محاوره‌ای است.
- (ب) **درست**: زیرا عامل هیچ‌گونه درک و اطلاعاتی از محیط ندارد و حسگرهای آن نمی‌توانند چیزی را حس کنند.
- (پ) **نادرست**:  $BFS$  کامل است اما  $DFS$  کامل نیست. هنگامی که از الگوریتم‌های ذکر شده روی یک گراف بی‌نهایت (بی‌حد و ضمنی) استفاده کنیم،  $BFS$  بالاخره جایگاه هدف (goal state) را پیدا می‌کند اما  $DFS$  ممکن است در بخش‌هایی از گراف که هیچ جایگاه هدفی (goal state) در آن‌ها وجود ندارد، پیش برود و در آن گم بشود و هرگز بازنگردد.
- (ت) **نادرست**: مزیت الگوریتم  $A^*$  این است که معمولاً  $node$ های کمتری را نسبت به  $BFS$  بررسی می‌کند اما اگر این را در نظر نگیریم،  $BFS$  سریع‌تر خواهد بود. اگر تابع هیورستیک مناسب نباشد یا گراف sparse یا کوچک باشد،  $BFS$  سریع‌تر خواهد بود. در صورتی که اگر جایگاه هدف در شاخه‌ی اول قرار داشته باشد،  $DFS$  سریع‌تر از  $A^*$  خواهد بود.
- (ث) **درست**: اگر تابع هیورستیک  $h(n)$  همواره هزینه را دست‌کم بگیرد ( $h(n) \leq h^*(n)$ )، آنگاه،  $A^*$  حتماً جواب بهینه را پیدا می‌کند.
- (ج) **نادرست**: در یک گراف، هنگامی که یک  $node$  را جست‌وجو می‌کنیم، دیگر هرگز آن را جست‌وجو نمی‌کنیم اما در یک درخت، ممکن است یک  $node$  چندین بار دیده و جست‌وجو شود که در آن صورت، جواب دیگر بهینه نخواهد بود.

### سؤال ۲:

- (آ) حالت ابتدایی مسئله (موقعیت آغازین در مارپیچ)  
Initial State:  $S_0$
- شیوه‌های مختلفی که که عامل می‌تواند تغییر موقعیت دهد.  
Actions:  $A$   
(حرکت به خانه‌های مجاور)
- شرط هدف (تابعی که مشخص می‌کند که آیا یک حالت به دست آمده در نتیجه‌ی زنجیره‌ای از اعمال، جواب مسئله هست یا خیر؟)  
Goal Condition:  $G$
- تابع هزینه (تابعی که یک مقدار عددی را به زنجیره‌ای از اعمال، نسبت می‌دهد).  
Cost Function:  $c$
- حالت ابتدایی به همراه مجموعه‌ی اعمال، فضای حالت یا همان  
State Space  
فضای جست‌وجو را تشکیل می‌دهد. (مجموعه‌ی تمام حالت‌هایی که با حرکت از حالت ابتدایی می‌توان به آن‌ها رسید).

حالت ابتدایی مسئله، مرکز مارپیچ است که ربات در آن قرار دارد و رو به شمال قرار دارد. ربات می‌تواند رو به شمال، جنوب، شرق و غرب قرار بگیرد و می‌تواند مسافت مشخصی را تحت کنترل کاربر آن حرکت کند و هنگامی که به دیوار

خیلی نزدیک شود، پیش از برخورد متوقف می‌شود. به این ترتیب ربات می‌تواند مسافت مشخصی را در جهت شمال، جنوب، غرب یا شرق حرکت کند. در نتیجه، ربات در هر جهتی که حرکت کند، یا قبل از برخورد به دیوار متوقف می‌شود یا همچنان در مارپیچ به پیش می‌رود. ربات می‌تواند ۵ عمل را انجام دهد؛ به شمال، جنوب، غرب یا شرق بچرخد یا در حال حرکت باشد. در واقع، این ربات می‌تواند در همان جهتی که رو به آن قرار دارد، حرکت کند تا وقتی که به دیوار برسد که در آن صورت، متوقف می‌شود. به این ترتیب، برای هر عملی که انجام دهد، دو حالت می‌تواند اتفاق بیفتد؛ یا ربات می‌تواند در همان جهتی که رو به آن قرار دارد، حرکت کند یا نمی‌تواند زیرا به دیوار رسیده است. چون ربات می‌تواند در ۴ جهت حرکت کند، برای فضایی با اندازه  $n$  حالت مختلف وجود دارد.

حالت ابتدایی،  $(in(start), face(N))$  است.

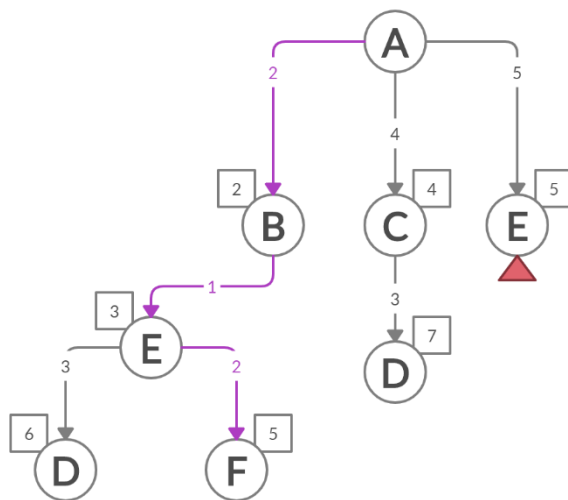
حالت بعدی به صورت  $(in(state), face(x)), x \in \{N, S, E, W\}$  خواهد بود.

هدف نیز خروج از مارپیچ است.

ب) اگر تقاطع چند راهرو، تشکیل‌دهنده‌ی بخشی از  $n$  مکانی باشد که آن حالت از آن تشکیل شده است، مثلاً  $(n - i)$ ، همچنان در هر حالت، تنها دو نتیجه ممکن است حاصل شود؛ یعنی  $(2(n - i))$  حالت. اما حالا ربات تنها می‌تواند در تقاطع‌ها به یکی از ۴ جهت گردش کند. این به معنای آن است که فضای حالت با اندازه‌ی  $4n$ ، حالا دارای اندازه‌ی  $4i$  است. در نتیجه، اندازه‌ی فضای حالت جدید برابر با  $2(n + i) = 2n + 2i = 4i + 2(n - i)$  خواهد بود.

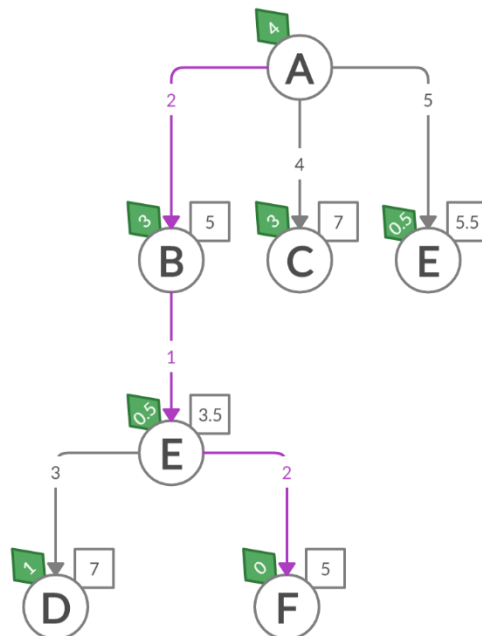
### سؤال ۳:

#### Uniform Cost Search



ترتیب گره‌های ملاقات شده:  $A \rightarrow B \rightarrow E \rightarrow C \rightarrow F$

#### $A^*$ (Euclidean Distance Heuristic)



ترتیب گره‌های ملاقات شده:  $A_4 \rightarrow B_5 \rightarrow E_{3.5} \rightarrow F_5$