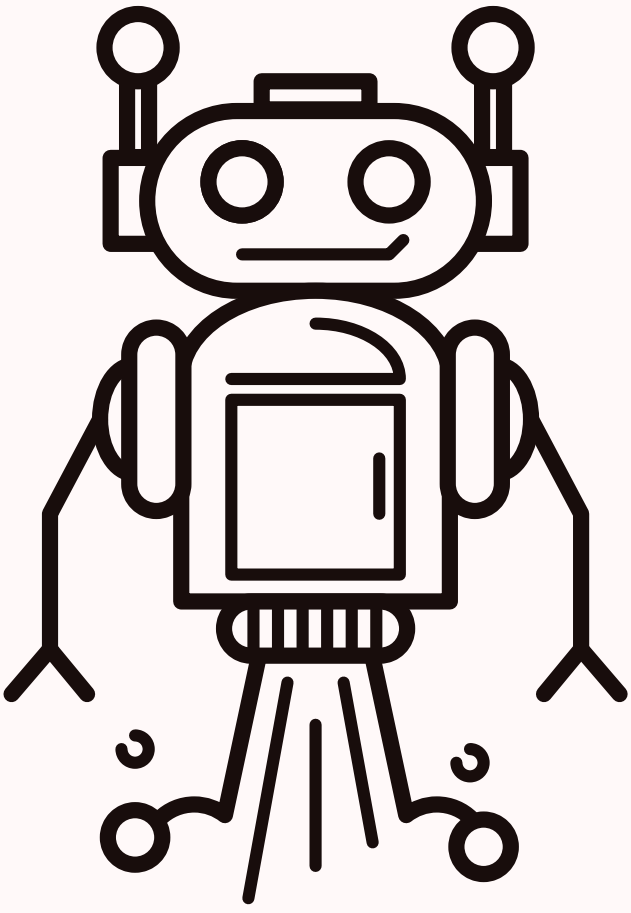
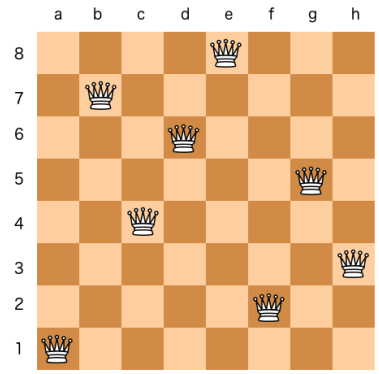
هوش مصنوعی

تمرین های نوشتنی

تمرین سوم

اسفند ۱۳۹۹ تا فرودین ۱۴۰۰





سئوال اول:

معرفی مسئله:

مسئله ۸ وزیر, توی این مسئله میخواهیم ۸ وزیر را در یک صفحه شطرنج ۸ در ۸ قرار بدیم به طوری که هیچ کدام نتونن هم دیگه رو بزنند.

پس به صورت کلی n وزیر داریم و صفحه n x n

جهت یادآوری: در شطرنج وزیر میتواند تمام خانه ها رو به صورت ضربدری و به علاوه طی کند پس باید طوری قرار داد که هیچ کدام هم دیگه روقطع نکنند.

توضیحات بیشتر در ویکی پدیا مسئله چند وزیر

۱- مسئله رو به عنوان یک مسئله جستجو ببنید که هر وضعیت کل صفحه شطرنج است. که هر مربع شامل شماره سطرو ستون و وجود وزیر در آن میباشد برای شروع ما اجازه میدیم که فضای وضعیتمون در هر حالتی و ترکیبی شامل n وزیر یا کمتر باشه.

وضعیت شروع را بنویسید:
هدف تست رو بنویسد:
تابع ساکسر رو بنویسید:

۲- فضای وضعیت چقدر بزرگ است با توجه به فرمول؟

۳- یکی از راه های کاهش سایز فضای وضعیت تغییر و کاهش خروجی تابع ساکس‌سور است. فرمول اون رو عوض کنید تا به فضای وضعیت بهینه تری برسید

۴- حالا با توجه به تغییرات سئوال سوم چقدر وضعیت در این فضای وضعیت وجود دارد؟

سئوال دوم:

برای اینجا از الگوریتم درخت برای جستجو استفاده میکنیم مگر اینکه در بخشی از سئوال به صورت خاص گفته باشه.

۱ - برای هر یک از الگوریتم های زیر نشانه دهید که ایا مسیر بازگشت داده شده بعد از ایجاد تغییر در درخت جستجو تضمین شده که یکسان با الگوریتم اصلاح نشده باشد. با توجه به اینکه هر یال غیر منفی هستن قبل از تغییر.
۱-۱- با توجه به اضافه کردن یک هزینه c > 0 به هر یال
۱-۲- ضرب یک عدد ثابت w > 0 به وزن هر یال

	حالت اول	حالت دوم
BFS		
DFS		
UCS		

برای جواب دادن جدولی مشابه روبرو بکشید
روبروی هر الگوریتم و حالت بنویسید بلی یا خیر

۲- برای این بخش از سئوال, دو الگوریتم به صورت مشابه تغریف شده و فقط در صورتی که در مسیر های مشابه گراف پیش روند و مسیر مشابه رو برمیگردوند (پس خروجیشون هم یکسان هست)**. با توجه به اینکه همه گراف ها به شکل جهت دار غیرمدور** (Directed acyclic graph) .
با توجه به اینکه ما دسترسی داریم به Cij که هزینه اجرای الگوریتم USC است که این هزینه Cij معادل اجرای BFS است.
سئوال اصلی این ایت که چطوری ما یک هزینه Cij جدید که USC رو معادل DFS اجرا کند چیست؟
(یکم دقیق تر بگم ما یک الگوریتم داریم USC میخواهیم از لحاظ هزینه اجرا مشابه DFS باشد کدام گزینه درست تر هستش با توجه به مثال قبل)

- ○ $c'_{ij} = 0$
- ○ $c'_{ij} = 1$
- ○ $c'_{ij} = c_{ij}$
- ○ $c'_{ij} = -c_{ij}$
- ○ $c'_{ij} = c_{ij} + \alpha$
- ○ امکان پذیر نیست

Q3. SpongeBob and Pacman (Search Formulation)

Recall that in Midterm 1, Pacman bought a car, was speeding in Pac-City, and SpongeBob wasn't able to catch him. Now Pacman has run out of gas, his car has stopped, and he is currently hiding out at an undisclosed location.

In this problem, you are on SpongeBob's side, tryin' to catch Pacman!

There are still p of SpongeBob's cars in the Pac-city of dimension m by n. In this problem, **all of SpongeBob's cars can move, with two distinct integer controls: throttle and steering, but Pacman has to stay stationary**. Spongebob's cars can control both the throttle and steering for each step. Once one of SpongeBob's cars takes an action which lands it in the same grid as Pacman, Pacman will be caught and the game ends.

Throttle: $t_i \in \{1, 0, -1\}$, corresponding to {Gas, Coast, Brake}. This controls the **speed** of the car by determining its acceleration. The integer chosen here will be added to the velocity for the next state. For example, if a SpongeBob car is currently driving at 5 grid/s and chooses Gas (1) it will be traveling at 6 grid/s in the next turn.
Steering: $s_i \in \{1, 0, -1\}$, corresponding to {Turn Left, Go Straight, Turn Right}. This controls the **direction** of the car. For example, if a SpongeBob car is facing North and chooses Turn Left, it will be facing West in the next turn.

(a) Suppose you can **only control 1 SpongeBob car**, and have absolutely no information about the remainder of $p - 1$ SpongeBob cars, or where Pacman has stopped to hide. Also, the SpongeBob cars can travel up to 6 grid/s so $0 \leq v \leq 6$ at all times.

(i) What is the **tightest upper bound** on the size of state space, if your goal is to use search to plan a sequence of actions that guarantees Pacman is caught, no matter where Pacman is hiding, or what actions other SpongeBob cars take. Please note that your state space representation must be able to represent **all** states in the search space.

(ii) What is the maximum branching factor? Your answer may contain integers, m, n.

(iii) Which algorithm(s) is/are guaranteed to return a path passing through all grid locations on the grid, if one exists?

☐ Depth First Tree Search ☐ Breadth First Tree Search
☐ Depth First Graph Search ☐ Breadth First Graph Search

(iv) Is Breadth First Graph Search guaranteed to return the path with the shortest number of **time steps**, if one exists?

○ Yes ○ No

(b) Now let's suppose you can control **all** p SpongeBob cars at the same time (and know all their locations), but you still have no information about where Pacman stopped to hide

(i) Now, you still want to search a sequence of actions such that the paths of p SpongeBob cars combined **pass through all $m * n$ grid locations**. Suppose the size of the state space in part (a) was N_1, and the size of the state space in this part is N_p. Please select the correct relationship between N_p and N_1.

○ $N_p = p * N_1$ ○ $N_p = p^{N_1}$ ○ $N_p = (N_1)^p$ ○ None of the above

(ii) Suppose the maximum branching factor in part (a) was b_1, and the maximum branching factor in this part is b_p. Please select the correct relationship between b_p and b_1.

○ $b_p = p * b_1$ ○ $b_p = p^{b_1}$ ○ $b_p = (b_1)^p$ ○ None of the above