DATE / / 4001098 43 SUBJECT تمرین ۱ - هوش مصنوعی

سوال ۱)

آ) نادرست.

تعریف ارائه شده، تنها تعریف یک agent است، اما برای rational بودن نیاز به خاصیت‌های زیر نیز دارد:

- The performance measure that defines the criterion of success
- The agent's prior knowledge of the environment
- The actions that the agent can perform
- The agent's percept sequence to date

ب) درست.

طبق تعریف rational بودن، این گزاره درست است.

ج) درست.

با استفاده از روش random restart به احتمال ۱ به یک مینیمم سراسری می‌رسیم. وی اگر از روش sideways move استفاده کنیم. تنها ۹۶٪ احتمال این است که به مینیمم سراسری برسیم.

د) درست.

DATE / / SUBJECT :

در الگوریتم Djikstra's Algorithm (یا Uniform-cost Search) با توجه به اینکه در هر رأس که باشیم، بهترین مسیر تا آن رأس را پیدا می‌کند. (به شرطی اینکه یال منفی نداشته باشیم و هر رأس تعداد مسیرهای یال داشته باشد.) در نتیجه این الگوریتم، الگوریتمی کامل و بهینه است.

ه) درست

در این الگوریتم، با احتمال نزدیک به ۱ می‌توانیم از دام بهینه‌های محلی فرار کنیم.

DATE / / SUBJECT :

سوال 2)

آ) حالت ها: هر حالت را به این صورت تعریف می کنیم که هر یک از این n کاوشگر را در کدام خانه از جدول nxn خود قرار می دهیم. به صورتی که State-space ما تمامی جایگشت های ممکن را حاصل شود.

کنش ها: برای یک کاوشگر کنش به این صورت تعریف می شود که یکی از کارهای راست، چپ، بالا، پایین، یا ثابت ماندن را انتخاب کند.

حال کنش ها به این صورت تعریف می شوند که تمامی این n کاوشگر یکی از کنش های ممکن را انتخاب کرده باشند. لازم به ذکر است که باید به روی این کنش ها، محدودیت را هم با توجه به حالت تعریف کنیم. به این معنا که اگر یک کاوشگر بر روی لبه صفحه باشد یا اینکه کنار یک کاوشگر باشد، به یقین بعضی از این کنش ها را نمی توانند انجام بدهد.

هزینه هر کنش: هر کاوشگر اگر در یکی از جهات حرکت کند هزینه 1 و اگر ثابت بماند. هزینه 0 دارد. در نتیجه هزینه هر کنش برابر است با جمع هزینه های مورد نیاز برای یک کاوشگر. که طبیعتا این هزینه به حالت بعدی و حالت قبلی بستگی دارد.

DATE / / SUBJECT :

هدف: همهٔ کاوشگرها به سطر پایانی برسند. به‌صوری که کاوشگری که در ابتدا در خانهٔ (1و1) قرار دارد در نهایت به خانهٔ $(n, n-i+1)$ برسد.

ب) $|State\ Space| = O\left(\binom{n^2}{n} \times n!\right) = O\left(\frac{n^{2!}}{(n^2-n)!}\right)$

برای حساب کردن تقریب استرلینگ این عبارت می‌توانیم از تقریب استرلینگ استفاده کنیم: $\left(x! \simeq \left(\frac{x}{e}\right)^x \sqrt{2\pi x}\right)$ که با میل دادن x به سمت بی‌نهایت این تقریب، تقریب خوبی می‌شود.

$$|State\ Space| = \frac{(n^2)!}{(n^2-n)!} \simeq \frac{\left(\frac{n^2}{e}\right)^{n^2} \times \sqrt{2\pi n^2}}{\left(\frac{n^2-n}{e}\right)^{n^2-n} \times \sqrt{2\pi(n^2-n)}} \simeq$$

$$\simeq \frac{\left(\frac{n^2}{e}\right)^{n^2} \times \sqrt{2\pi n^2}}{\left(\frac{n^2}{e}\right)^{n^2-n} \times \sqrt{2\pi n^2}} \simeq \left(\frac{n^2}{e}\right)^n \Rightarrow |State\ Space| = O\left((n^2)^n\right)$$

ج) کنش‌های قابل قبول برای یک کاوشگر برابر با 5 حالت است. در هر مرحله، هر کاوشگر باید یک کنش انجام دهد که در بدترین حالت، تمامی این کاوشگرها امکان حرکت دارند. که با توجه به قانون ضرب

$$Branching\text{-}factor = 5^n$$

DATE / / SUBJECT :

د) یک تابع اکتشافی قابل قبول برای یک کاوشگر مجله برابر با فاصله منهتن است به این صورت که

$$h_i(x_i, y_i) = |n - i + 1 - x_i| + |n - y_i|$$

دلیل اینکه این تابع، یک تابع قابل قبول است این است که:

کوتاه ترین فاصله تا مقصد، برابر با فاصله منهتن است. و اگر فاصله تا مقصد برابر با $d_i(x_i, y_i)$ باشد، حتماً داریم که $h_i(x_i, y_i) < d_i(x_i, y_i)$.

در صورتی که کاوشگرهای دیگر در سیاره نیز قرار داشته باشند، فاصله یک کاوشگر تا مقصد بیشتر می شود (بقیه کاوشگرها مانند مانع عمل می کنند) در این شرایط، فاصله همیشه $d_i(x_i, y_i) + \epsilon_i(x_i, y_i)$ می شود که بدین است که

$$h_i(x_i, y_i) < d_i(x_i, y_i) + \epsilon_i(x_i, y_i)$$

و این تابع اکتشافی یک تابع admissible می باشد.

DATE / / SUBJECT :

ه) در نظر بگیرید که برای مسئله اصلی تعداد گام تا رسیدن به جواب بهینه برابر با d' و حدسی که از آن داریم (heuristic مربوط به آن) برابر با h' است.

بدیهی است که مقدار d' بیشتر از $\max\{d_1, d_2, \dots, d_n\}$ است.

زیرا با یک گام ماکزیمم حداکثر به ماکزیمم برسد و در شرایط اصلی مسئله، ممکن است که این گام در مسیر خود به مانع‌هایی برخورد کند.

در نتیجه داریم که $d' \geq \max\{d_1, d_2, \dots, d_n\}$ است.

~~در حالت کلی داریم که $d_i(x_i, y_i) \leq h_i(x_i, y_i)$~~

$$\left.\begin{array}{l} \forall i: d_i(x_i, y_i) \geq h_i(x_i, y_i) \\ \forall i: \min\{h_1, h_2, \dots, h_n\} \leq h_i \end{array}\right\} \Rightarrow \min\{h_1, h_2, \dots, h_n\} \leq d_i(x_i, y_i)$$

$$\Rightarrow \min\{h_1, h_2, \dots, h_n\} \leq d'$$

در نتیجه $\min\{h_1, h_2, \dots, h_n\}$ یک heuristic قابل قبول است.

$$\max\{h_1, h_2, \dots, h_n\} \leq \max\{d_1, d_2, \dots, d_n\} \leq d'$$

در نتیجه $\max\{h_1, h_2, \dots, h_n\}$ نیز یک heuristic قابل قبول است.

$$h_i \leq \max\{h_1, h_2, \dots, h_n\} \Rightarrow \sum h_i \leq n \max\{h_1, h_2, \dots, h_n\}$$

$$\Rightarrow \frac{\sum h_i}{n} \leq \max\{h_1, h_2, \dots, h_n\} \leq d'$$

در نتیجه $\frac{\sum h_i}{n}$ نیز heuristic قابل قبول است.

DATE / / SUBJECT :

حال با یک مثال نقض هر سه تابع $\max\{h_1,\dots,h_n\}$، $\min\{h_1,\dots,h_n\}$ و $\sum h_i$ را رد می‌کنیم (قابل قبول نیستند)

حالتی را در نظر بگیرید که کاوشگری که ابتدا در خانه (1و1) قرار داشته، در حال حاضر در خانه (1-nو1+i-n) قرار داشته باشد.

حال در یک مرحله تمامی این کاوشگرها می‌توانند با یک گام حرکت کردن به خانه (nو1+i-n) برسند که در واقع همان State هدف ماست.

در نتیجه این‌که است اما h_iها است که

$$\min\{h_1,\dots,h_n\} = \max\{h_1,\dots,h_n\} = \sum h_i = n \Rightarrow$$

در نتیجه این دو تابع، تابع‌های قابل قبولی نیستند.

در نهایت به صورت خلاصه:

غیر قابل قبول ← $\max\{h_1,\dots,h_n\}$ ، $\min\{h_1,\dots,h_n\}$ ، $\sum h_i$

قابل قبول ← $\min\{h_1,\dots,h_n\}$ ، $\max\{h_1,\dots,h_n\}$ ، $\frac{\sum h_i}{n}$

و)

و-آ) حالت ها مانند قسمت قبلی هستند.

کنش ها در این مسئله بدین صورت است که تنها یکی از این کاوشگرها امکان حرکت دارد. {چپ، راست، بالا، پایین} و اگر یکی از این کاوشگرها حرکت کند، بقیه نبایستی که حرکت کنند. (همه نیز می توانند حرکت کنند)

هزینه هر کنش نیز می تواند صفر یا 1 باشد، وابسته به اینکه کاوشگری حرکت می کند یا نه.

هدف نیز به این صورت است که همه در خانه مقصد قرار گرفته باشند، به صورتی که کاوشگر i ام، نفر i ام به هدف خود رسیده باشد.

و-ب)

در State Space حالاتی که $(k < i)$ و k زودتر از i برسد را در نظر نمی گیرم.

$4n+1$

و-ج)

و-ب) State Space نیز مانند قسمت قبل است.

$$\binom{n^2}{n} n! = \frac{(n^2)!}{(n^2-n)!} = O((n^2)^n)$$

DATE / / SUBJECT :

سوال 3)

آ) برای بررسی پذیرفتنی بودن باید چک کنیم که برای دو Node مختلف آیا $|h_A - h_B| \leq Cost_{A-B}$ است یا نه. (به شرط اینکه همسایه باشند)

برای h_1 داریم:

$|h_{1A} - h_{1C}| = 1,5 \leq 4$ $|h_{1A} - h_{1B}| = 0,5 \leq 1$ ~~$|h_{1A} - h_{1B}|$~~

$|h_{1C} - h_{1D}| = 1 \leq 3$ $|h_{1B} - h_{1D}| = 2 \leq 5$ $|h_{1B} - h_{1C}| = 1 \leq 1$

$|h_{1D} - h_{1G}| = 7 \leq 9$ $|h_{1D} - h_{1F}| = 3 \leq 3$ $|h_{1D} - h_{1E}| = 5,5 \leq 8$

$|h_{1F} - h_{1G}| = 4 \leq 5$ $|h_{1E} - h_{1G}| = 1,5 \leq 2$

در نتیجه تابع h_1، یک تابع پذیرفتنی است.

برای h_2 بین A و B اختلاف h_2 برابر 2 است اما Cost بین این دو Node برابر 1 است. در نتیجه تابع h_2 نیز یک تابع پذیرفتنی است.

برای چک کردن قابل قبول بودن heuristic ها، نیز تنها باید چک کنیم که مقدار h برای هر Node کوچکتر یا مساوی Cost تا هدف باشد که هر دو h_1 و h_2 قابل قبول هستند

DATE / / SUBJECT :

(ب)

	A-B-D-G	A-C-D-G	A-B-C-D-F-G
جست وجوی عمق اول	✓	✓	✓
" سطح " " "	✓	✓	✗
" " هزینه یکنواخت	✗	✗	✓
h_1 ؛ A^*	✗	✗	✓
h_2 ؛ A^*	✗	✗	✓

A^* ؛ h_1:

$h(n) = g(n) + h(n)$

در هر مرحله برای رأس ها همسایه مقدار $h(n)$ را حساب می‌کنیم و مقدار کمتر را انتخاب می‌کنیم

A { B: $h(n) = 9+1$ ✓ ; C: $h(n) = 8+4$ ✗ }

B { C: $h(n) = 1+5+7$ ✗ ; D: $h(n) = 1+1+8$ ✓ }

C { D: $h(n) = 2+3+7$ ✓ }

D: { E: $5+8+1,5$ * ; F: $5+3+4$ ✓ ; G: $5+9+0$ ✗ }

$A \to B \to C \to D \to F \to G$ ⟵ مسیر قابل قبول

A^* ؛ h_2 نیز مانند بالا عمل می‌کنیم و در نهایت مجدد

مسیر A-B-C-D-F-G قبول می‌شود.

DATE / / SUBJECT :

ج) برای قابل قبول بودن $h_3(B)$ باید این مقدار کوچکتر از k min فاصله تا هدف باشد. $\min\{distance\} = 12$

$$h_3(B) \leq 12$$

د) برای یکنوا بودن این تابع باید $|h_3(B) - h_3(c)| \leq cost(c)$ برای تمامی همسایه‌های B باشد.

$$A: |h_3(B) - 10| \leq 1 \Rightarrow 9 \leq h_3(B) \leq 11$$
$$C: |h_3(B) - 9| \leq 1 \Rightarrow 8 \leq h_3(B) \leq 10$$
$$D: |h_3(B) - 7| \leq 5 \Rightarrow 2 \leq h_3(B) \leq 12$$
$$\Rightarrow \quad 9 \leq h_3(B) \leq 10$$

ه)

$$A: \begin{cases} B: 1+h_3(B) \\ C: 4+9 \checkmark \end{cases} \Rightarrow h_3(B) \geq 12$$

$$C: \begin{cases} B: 4+1+h_3(B) \checkmark \\ D: 4+3+7 \times \end{cases} \Rightarrow h_3(B) \leq 9$$

در نتیجه همچین چیزی ممکن نمی‌باشد.

DATE / / SUBJECT :

سوال ۶)

آ) در این مسئله، جایگذرد می خواهد با این n کلمه یک جمله بسازد.
در نتیجه فضای مسئله به صورت یک جایگشت از n کلمه است.

$|State\ Space| = n!$

ب) دو همسایه در این مسئله به این صورت هستند که جایگاه تنها دو کلمه در این جمله ها برعکس همدیگر باشند و $n-2$ کلمه دیگر در یک جایگاه قرار داشته باشند.

مثال: (1) هرچه می کنم خواهش زودتر من را آزاد کنید
(2) خواهش آزاد هرچه زودتر من را می کنم کنید.

ج) اگر تنها از الگوریتم Hill Climbing استفاده کنیم تضمین به این که به جواب درست برسیم نداریم و تنها به احتمال مانند p به جواب می رسیم.
اگر از Random Walk استفاده کنیم، باز هم تضمین برای اینکه به جواب برسیم نداریم و تنها احتمال به جواب رسیدن را مقدار خوبی اضافه می کنیم.
اما با استفاده از Random Restart با احتمال نزدیک به (1) به جواب می رسیم.

DATE / / SUBJECT :

د) mutation:

برای mutation می‌توان به این صورت عمل کرد که جایگاه دو کلمه را با هم عوض کنیم (در واقع از یک Shake به یک Shake مجاور برویم)

Crossover:

برای این قسمت می‌توان از روشی به نام order-Crossover استفاده کرد.

این روش به این صورت عمل می‌کند که یک substring از parent1 را انتخاب می‌کنیم و آن substring را در همان جایگاه (offspring) نگه می‌داریم

سپس کلماتی که در این substring قرار دارند را از parent2 حذف می‌کنیم. حال در offspring (جمله دومی که در حال ساخت آن هستیم) کلماتی که در parent2 باقی مانده‌اند را به ترتیب در جایگاه‌های خالی offspring قرار می‌دهیم.

مثال: substring

parent1: زودتر کنیه هرچه می‌نم آزاد را من خواهش

parent2: زودتر خواهش من هرچه را کنیه می‌نم آزاد

offspring: زودتر خواهش من می‌نم آزاد را هرچه کنیه

سوال 5 /

ابتدا برای این مسئله اثبات می کنیم که

$$\nabla(x^T B x) = (B + B^T)x$$

$$f = x^T B x \xrightarrow{\text{By Definition}} f = \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} x_i x_j \rightsquigarrow$$

$$\rightsquigarrow \frac{\partial f}{\partial x_k} = \sum_{j=1}^{n} b_{kj} x_j + \sum_{i=1}^{n} b_{ik} x_i = (Bx)_k + (B^T x)_k \Rightarrow$$

$$\Rightarrow \nabla f = (B + B^T)x$$

* $f: \mathbb{R}^n \to \mathbb{R}, f(x) = \|x\|_2^2 = x^T x \rightsquigarrow \nabla f = (I + I^T)x = 2x$ (i

* $f: \mathbb{R}^n \to \mathbb{R}, f(x) = \|Ax\|_2^2 = x^T A^T A x \quad (Ax)^T Ax = x^T A^T Ax \Rightarrow$

$$\Rightarrow \nabla f = (A^T A + (A^T A)^T)x = 2A^T Ax$$

* $f: \mathbb{R}^n \to \mathbb{R}, f(x) = \|Ax - b\|_2^2 + \gamma\|x\|_2^2 = (Ax-b)^T(Ax-b) + \gamma x^T x =$

$$= (x^T A^T - b^T)(Ax - b) + \gamma x^T x = x^T A^T Ax - b^T Ax - x^T A^T b + b^T b + 2\gamma x^T x \Rightarrow$$

$$\Rightarrow \nabla f = 2A^T Ax - (b^T A)^T - A^T b + 2\gamma x =$$

$$= 2A^T(Ax - b) + 2\gamma x$$

$$c^T x = \sum_{j=1}^{n} c_j x_j \rightsquigarrow \frac{\partial}{\partial x_k}(c^T x) = c_k \Rightarrow$$ (ب.ن.ه

$$\nabla(c^T x) =$$

$$\Rightarrow \nabla(c^T x) = C$$

DATE / / SUBJECT :

ب) در الگوریتم gradient descent در هر مرحله x را به این صورت تغییر می دهیم که

$$x = x - \alpha \nabla f_{(x)}$$

در زمانی که f تابع در یک نقطه min گیر کند، x تغییری پیدا نمی کند

حال برای پیدا کردن این نقطه min گرادیان تابع را برابر با صفر می گذاریم.

$f_{(x)} = \|x\|_2^2 \rightsquigarrow \nabla f_{(x)} = 0 \Rightarrow 2x = 0 \Rightarrow x = 0$

$f_{(x)} = \|Ax\|_2^2 \rightsquigarrow \nabla f_{(x)} = 2A^TAx = 0 \Rightarrow x = 0$

$f_{(x)} = \|Ax-b\|_2^2 + \gamma\|x\|_2^2 \Rightarrow \nabla f_{(x)} = 2A^T(Ax-b) + 2\gamma x = 0 \Rightarrow$

$\Rightarrow (2A^TA + \gamma)x = A^Tb \Rightarrow$

$\Rightarrow x = (A^TA + \gamma)^{-1}A^Tb$

لازم به ذکر است برای اینکه این تابع به min برسد از $x = x - \alpha \nabla f_{(x)}$ استفاده می کنیم. به صورتی که در هر مرحله به اندازه $\alpha \nabla f_{(x)}$ مقدار x ما تغییر می کند. که در آن α را step است و این کار را به اندازه ای تغییر می دهیم تا در min که در بالا به دست آوردیم، گیر نکنیم.

DATE / / SUBJECT :

ج)

$$h(x) = \|x\|_2^2 = \sum_{k=1}^{n} x_k^2$$

$$\frac{\partial^2 h(x)}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}(2x_j) = 0$$

$$\frac{\partial^2 h(x)}{\partial x_i^2} \Rightarrow \frac{\partial}{\partial x_i}(2x_i) = 2$$

$\Rightarrow$ Hessian is positive

با توجه به اینکه ماتریس Hessian این تابع مثبت است

در نتیجه این تابع، تابعی محدب است.

د)

$$g(\lambda x + (1-\lambda)y) = h(A(\lambda x + (1-\lambda)y) - b) = h(\lambda(Ax-b) + (1-\lambda)(Ay-b))$$

$$\leq \lambda h(Ax-b) + (1-\lambda) h(Ay-b) = \lambda g(x) + (1-\lambda) g(y) \Rightarrow$$

$$\Rightarrow g(\lambda x + (1-\lambda)y) \leq \lambda g(x) + (1-\lambda) g(y)$$

↳ Convex

ه)

$$h(x) = \|x\|_2^2 \rightarrow \text{Convex}$$

$$g(x) = \|Ax-b\|_2^2 = h(Ax-b) \rightarrow \text{Convex}$$