به نام خدا

تمرین تئوری ۱ | مبانی هوش مصنوعی

مهدی شاهینی (۹۹۲۳۰۴۰)

۱- الف	کارآیی	محیط وظیفه	عملگرها	سنسورها
سیستم تشخیص نفوذ شبکه	تشخیص پکت‌ها و پروتکل‌های ارتباطی مخرب و تمیز دادن آن‌ها نسبت به موارد صحیح و غیر مخرب تحت شبکه، حفظ سرعت تشخیص متناسب سرعت شبکه (fail نشدن در صورت ارسال پکت با سرعت بالا یا رمزگذاری پیچیده)	شبکه‌های محلی و گلوبال اینترنت و اینترانت و هر نوع شبکه ارتباطی میان تجهیزات عموما دیجیتال یا آنالوگ داره گیرنده-فرستنده آنالوگ و دیجیتال (روتر، سرور، کلاینت و ...)، پروتکل‌های مرسوم شبکه	سیستم‌های هشدار دهنده (آلارت): اطلاع رسانی ایرادات و موارد مخرب به صورت‌های مختلف یا مانع کننده (بلاکر): بستن و قفل کردن شبکه، روتر، سرور و ...	تجهیز کنترل ترافیک شبکه: تشخیص الگوها و سرعت ارسال آن‌ها رجیسترهای ذخیره کننده تاریخچه شبکه و پتک‌ها (log) کنترلر سرور، هاست و ... جهت جلوگیری از آسیب یا نفوذ کابران و پتک‌ها مخرب
ربات معامله‌گر بورس	پیش‌بینی دقیق بازار سهام، بهینه کردن میزان سود و ضرر در افت و خیزهای شاخص بورس، خرید و فروش به موقع در زمان صحیح، مدیریت ریسک و نرخ بازده	شاخص‌های بازار سرمایه و بورس، مراجع اطلاع رسانی اخبار رسمی شرکت‌های صاحب سهام از مراجع اطلاع رسانی، حساب‌های بانکی خریدار (اگر بنا به تبدیل پول به سرمایه باشد.)	الگوریتم‌های یادگیری ماشین و هوش مصنوعی تحلیل کننده، سیستم‌های تصمیم‌گیری و نحوه اتصال و فرمان‌دهی آن‌ها به حساب و اکانت صاحب ربات	ابزارهای جمع‌آوری اطلاعات و داده‌ها از منابع، چارت‌ها و رسانه‌ها جهت غنی کردن عملگرها
قفسه‌های متحرک انبارهای آمازون	قراردادن کالاها در محل صحیح در سریع‌ترین زمان ممکن با کمترین تداخل برای سایر ربات‌ها، رسیدن به محل اخذ بار در سریع‌ترین زمان و دقیق‌ترین مسیر	محیط انبار، قفسه‌ها و راهروها، خودروهای باری و ربات‌های دیگر حمل و نقل، اپراتورهای انسانی دیگر	چرخ‌ها و بازوهای ربات	سخت‌افزاری: فاصله، لیزر، التراسونیک، پارامترهای موتور و جهت و مکان‌یابی نرم‌افزاری: مکان مطلوب که به ربات داده می‌شود

۱- ب	تحلیل محیط
سیستم تشخیص نفوذ شبکه	اگر فرض شود کل شبکه در لحظه مانتیور شود کاملا قابل مشاهده، چند عامله (فرستنده پکت)، تصادفی (معلوم نیست در لحظه بعد چه قدر ارسال شود)، پیوسته، ترتیبی و داینامیک (ممکن است حجم زیادی پکت ارسال شود که بررسی آن‌ها زمانبر است پس ت و د)، شناخته
ربات معامله‌گر بورس	اگر اتفاقات و ... را در نظر بگیریم نیمه مشاهده پذیر، چندعامله، باتوجه به شرایط محیط غیرقطعی، پیوسته، ترتیبی، داینامیک، تا حدی ناشناخته (می‌تواند کم کم شناخته شود.)
قفسه‌های متحرک انبارهای آمازون	نیمه مشاهده‌پذیر (جزئی)، تک عامله (عامل همکار یا متخاصم نداریم)، بسته به تعریف مسئله می‌تواند تصادفی یا استراتژیک باشد. ترتیبی، داینامیک، پیوسته

۲-

برای فرموله کردن مسئله باید موارد زیر مشخص شود:

حالت‌ها، حالت شروع، فعالیت‌ها، مدل گذار، آزمون هدف، هزینه مسیر

در حل سوال سعی کرده‌ام پاسخ الف و ب را یکجا و جامع بدهم.

حالت‌ها:

به فرم ریاضی:

$$i\text{: } left\ side, j\text{: } right\ side \rightarrow \binom{4}{i} = \binom{4}{j = 4 - i}$$

که یعنی اگر حالت یک طرف را بشماریم حالات طرف دیگر مکمل آن است و نیازی به شمردن ندارد.

پس داریم:

$$\sum_{i=0}^{4} \binom{4}{i}$$

به این صورت می‌توان کلیه حالات موجود را با فرض عملگرهای زیر در دو طرف نشان داد. پس خواهیم داشت:

A for t=1 min, B for t=2 min, C for t=5 min & D for t=10 min so we will have:

L<-,-,-,-> , R<-,-,-,->

مثلا در حالت شروع:

L<A, B, C, D> ,R<>

آزمون هدف این است که به استیت هدف زیر رسیده باشند و البته قید زمان رعایت شده باشد:

L<> , R<A, B, C, D> in $t \leq 17$

فعالیت‌ها به صورت حرکت از یک سمت به سمت دیگر با هزینه کندترین عامل است یعنی:

$$actions\text{: } \xrightarrow{\max(t_i,t_j)}, \xleftarrow{\max(t_i,t_j)}$$

فعالیت‌ها جا به جایی ۱ یا ۲ عامل از یک سمت به سمت دیگر است که در مدت زمان کندترین عامل صورت می‌گیرد و هزینه آن عمل نیز همان مدت زمان محسوب می‌گردد. مسلم است برای رسیدن به هدف باید در حرکت از سمت مبدا به مقصد دوتایی و از مقصد به مبدا تنهایی حرکت کنند!

برای مدل گذار می‌توان به صورت زیر نمایش داد:

$$(L < 1, 2, or\ more\ agents >, R < other\ agents >)_{k-1} \xrightarrow{\max(t_i,t_j)} (L < A\ couple\ of\ less >, R < A\ couple\ of\ more >)_k$$

$$(L < 1\ more\ agents >, R < 1\ agenet\ less >)_k \xleftarrow{t_i} (L < other\ agents >, R < 1, 2, or\ more\ agents >)_{k-1}$$

نوع فعالیت به صورت فلش، هزینه فعالیت بالانویس فلش و استیت قبلی بعدی سمت ابتدا و انتهای فلش قرار می‌گیرند.

مثلا خواهیم داشت:

$$L < A, B, C, D >, R <> \xrightarrow{2} L < C, D >, R < A, B > then\ L < A, C, D >, R < B > \xleftarrow{1} L < C, D >, R < A, B > then\ L < A, C, D > R < B > \xrightarrow{10} L < A >, R < B, C, D >$$

۳- الف)

۱- ابعاد مپ، MN است پس عامل در MN مکان مختلف میتوان حضور داشته باشد. همچنین اگر هر مپ خاص فرض شود مثلا این مپ، پس هر روح می‌تواند باشد می‌تواند نابود شده باشد پس 2^G حالت برای هر روح‌ها وجود دارد. درنهایت: $MN2^G$

۲- اگر تنها یک روح وجود داشت بله می‌توانست باشد اما وقتی چند روح وجود دارد خیر! زیرا اگر حرکت به صورت مبدا، روح۱، روح۲، ...، روح n باشد آن‌گاه منطقی نیست که فاصله روح‌ها نیز از یکدیگر همواره بیشتر از فاصله پکمن از تک تک روح‌ها باشد پس ناسازگار و غیر قابل قبول است.

۳- حتی اگر حالت قبل به صورت جمع کمترین فاصله منهتنی باشد (ضرب تعداد روح‌ها در کمترین فاصله منهتنی) باز هم منطقی نیست! زیرا تضمینی وجود ندارد که کران پایین فاصله منهتنی بزرگتر از فاصله پکمن از روح n+1 است زمانی که روح n ام را میخورد! پس غیرقابل قبول و ناسازگار است.

برای پاسخ سوال ۲ و ۳ اولین مثال ساده این است که روح‌ها در یک ردیف و متوالی هستند!

۴- همانطور که در مثال هم گفته شد؛ کمترین میزان حرکت زمانی است که ما حداقل یک خانه خرج کنیم تا به آن برسیم یعنی روح‌ها در خانه‌ها بهم چسبیده قرار دارند. پس این پیشنهاد سازگار و قابل قبول است.

۳- ب)

۱- اگر توزیع مکان روح‌ها و پکمن‌ها یکسان نباشد مثلا تعدادی از روح‌ها بسیار نزدیک باشند بنابراین احتمالا اینکه تعدادی از پکمن‌ها پیش از رسیدن سایرین همه روح‌ها را بخورند وجود دارد و چنین هیورستیکی ناسازگار و غیرقابل قبول است.

۲- اگر باز هم مثال قبل را در نظر بگیریم هر پکمن حداقل باید فاصله تا نزدیک‌ترین روح را بپیماید پس منطقی است چنین هیورستیکی را در نظر بگیریم و قابل قبول و سازگار است.

۳- خیر، نه سازگار و نه قابل قبول است. اگر روح‌ها چسبیده به پکمن‌ها باشند آنگاه در استیت بعدی همه‌ی روح‌ها خورده شده‌اند و تنها در یک حرکت این صورت گرفته پس منطقی نیست.

۴- دقیقا اگر مثال قبل را فرض کنیم حداقل تعداد حرکات ۱ است بنابراین چنین هیورستیکی که حالت ایده‌آلی را در نظر گرفته قابل قبول و سازگار است.

۴- الف)

$$for\ h_1:$$

$$A_{9.5} \to B_{10}, C_{12} \xrightarrow{B_{10}} C_{10}, \cancel{C_{12}}, D_{13} \xrightarrow{C_{10}} D_{12}, \cancel{D_{13}} \xrightarrow{D_{12}} F_{12}, E_{13}, G_{14} \xrightarrow{F_{12}} E_{13}, G_{13}, \cancel{G_{14}} \xrightarrow{E_{13}} G_{13}, \cancel{G_{15}}$$

$$A, B, C, D, F, G\ with\ real\ cost\ 13$$

ابتدا ملزم به انتخاب A هستیم. سپس نودهای اکسپند شده B و C هستند. نود B هزینه کمتری دارد و انتخاب می‌شود. با انتخاب B نود C با وزن کمتر بروز رسانی می‌شود و نود D نیز اکسپند می‌شود. با انتخاب نود C، نود D نیز با هزینه کمتر بروز رسانی می‌شود. با انتخاب D، نودهای F و E و G اکسپند می‌شوند. با انتخاب نود F، نود هدف با هزینه کمتر بروز رسانی می‌شود. در اینجا چون ملاک هیورستیک قابل قبول می‌باشد پس هزینه واقعی نود E قطعا بزرگتر مساوی G_{13} است و میتوان هزینه را ۱۳ اعلام کرد حال اگر E را اکسپند کنیم G_{15} بدست می‌آید که از قبل هم می‌دانیم بدیهی بود هزینه بیشتری از آن مسیر خواهد داشت.

$$for\ h_2:$$

$$A_{10} \to B_{13}, C_{14} \xrightarrow{B_{13}} C_{12}, \cancel{C_{14}}, D_{14} \xrightarrow{C_{12}} D_{13}, \cancel{D_{14}} \xrightarrow{D_{13}} F_{12.5}, E_{12+2}, G_{14} \xrightarrow{F_{12.5}} E_{12+2}, G_{13}, \cancel{G_{14}}, \xrightarrow{E_{12+2}} G_{13}, \cancel{G_{15}}$$

$$A, B, C, D, F, G\ with\ real\ cost\ 13$$

مشابه قسمت قبل عمل می‌کنیم اما چون هیورستیک ناسازگار است و **شرط زیر برقرار <u>نیست</u>**. لذا احتمال خطا وجود دارد؛

$$h(E) - h(G) = h(E) \le cost(E \to G)$$

پس در انتها الزامی است نود E را نیز اکسپند کنیم و نمیتوان مانند بخش قبل زودتر پیش‌بینی کرد زیرا ممکن است هزینه واقعی رفتن از نود E به نود G کمتر از ۱ باشد در نتیجه هزینه رسیدن به هدف بین ۱۲ تا ۱۳ می‌شود و آن مسیر کم هزینه‌ترین خواهد بود لذا پس از اکسپند کردن متوجه شدیم که همان مسیر با هزینه ۱۳ قابل قبول بود.

۴- ب) مفروضات (شرط قابل قبول بودن و شرط سازگاری)؛ برای هر نود X و Y داریم:

$$h(X) \le real\ cost\ of\ X \to G$$

$$h(X) - h(Y) \le real\ cost\ of\ X \to Y$$

۱- هزینه کل مسیر ۱۳ و هزینه مسیر B تا G برابر ۱۲=۱-۱۳ است پس هیورستیک B باید کوچکتر ۱۲ باشد.

$$0 \le h(B) \le 12$$

۲- شرط سازگاری را بین هر نقطه و نقطه B می‌نویسیم و از قیود حاصل نتیجه‌گیری می‌کنیم.

$$h(A) - h(B) \le c(A, B) \to h(A) - c(A, B) \le h(B)$$

$$h(B) - h(A) \le c(B, A) \to h(B) \le h(A) + c(B, A)$$

$$h(C) - h(B) \le c(C, B) \to h(C) - c(C, B) \le h(B)$$

$$h(B) - h(C) \le c(B,C) \to h(B) \le h(C) + c(B,C)$$

$$h(D) - h(B) \le c(D,B) \to h(D) - c(D,B) \le h(B)$$

$$h(B) - h(D) \le c(B,D) \to h(B) \le h(D) + c(B,D)$$

شرط اول بزرگتر از ۹ ، شرط دوم بزرگتر از ۸ و شرط پنجم بزرگتر از ۲ را نتیجه می‌دهند.

شرط دوم کوچکتر از ۱۱ ، شرط سوم کوچکتر از ۱۰ و شرط ششم کوچکتر از ۱۲ را نتیجه می‌دهند.

پس هیورستیک سازگار برای B برابر است با:

$$9 \le h(B) \le 10$$

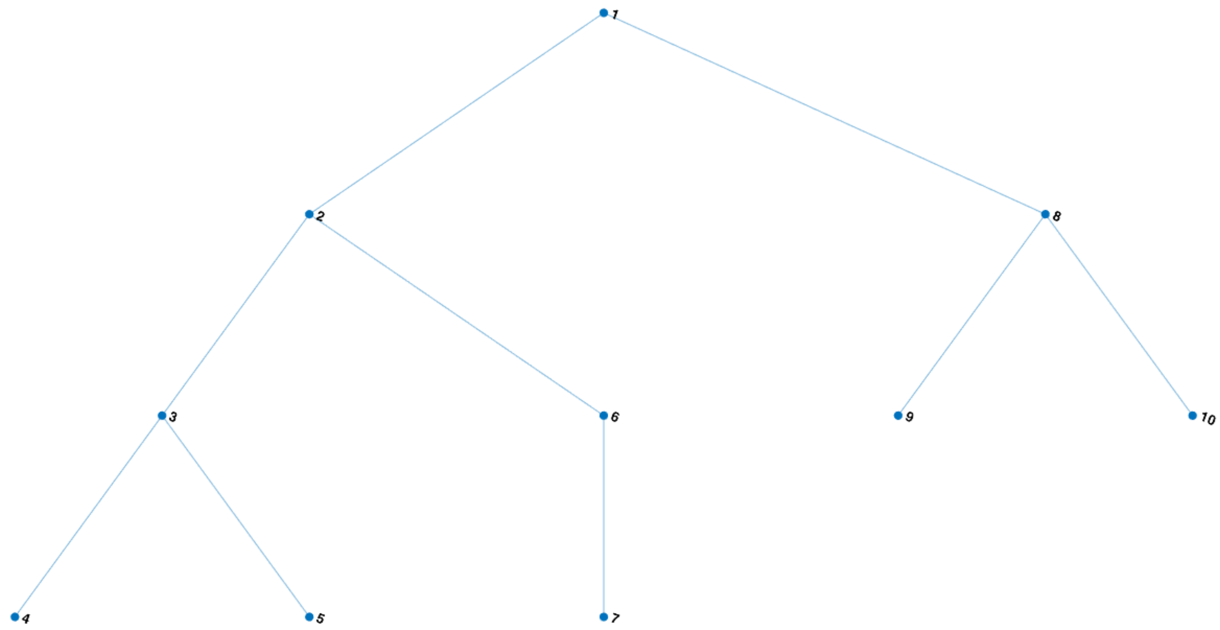
۳- پس از اینکه ناچارا A انتخاب شد نودهای B و C به ترتیب با هزینه‌های $h(B) + 1$ و ۱۳ اکسپند می‌شوند لذا هیورستیک B کمتر از ۱۲ است. پس از انتخاب C نودهای B و D با هزینه‌های $h(B) + 5$ و ۱۴ اکسپند می‌شوند. پس هیورستیک B کمتر از ۹ است.

۴- ج) اگر از هیورستیکی که کاملا قابل قبول باشد استفاده کنیم در صورتی که از جست و جوی گرافی استفاده کنیم الزامی است هیورستیک سازگار نیز باشد اما اگر از جست‌وجوی درختی استفاده کنیم قابل قبول بودن کفایت می‌کند.

۴- د) هیورستیک قابل قبول برای به حاشیه بردن مسیرهای با هزینه بالاتر سبب کندی اجرا می‌شود اگر هیورستیک غیرقابل قبول سبب خطا در مسیر نهایی نشود سرعت اجرای الگوریتم را می‌تواند بالا ببرد.

۵-

مثلا اگر هدف در عمیق‌ترین لایه باشد یا در لایه‌ای نسبتا عمیق و در سمت چپ نمودار باشد؛ آنگاه الگوریتم *DFS* سریعتر از *IDS* خواهد بود. مثلا در مثال زیر، اگر هدف نود ۷ باشد؛ آنگاه در بررسی آخرین لایه به جواب خواهیم رسید اما اگر *DFS* بزنیم سریع‌تر خواهیم رسید. پیچیدگی زمانی هر دو از $O(b^d)$ است اما تعداد محاسبات بیشتری *IDS* خواهد داشت.



BFS, Graph		BFS, Tree		DFS, Graph		DFS, Tree	
Frontier	**Explored**	**Frontier**	**Explored**	**Frontier**	**Explored**	**Frontier**	**Explored**
S		S		S		S	
A, B, D	S	A, B, D	S	A, B, D	S	A, B, D	S
B, D, C	S, A	B	S, A	C, B, D	S, A	C, B, D	S, A
D, C, ~~D~~	S, A, B	D, C, D	S, A, B	~~D~~, G, B, D	S, A, C	D, G, B, D	S, A, C
G, C	S, A, B, D	G, C, D	S, A, B, D				
The Shortest Path: 4 S, D, G		The Shortest Path: 4 S, D, G		The Shortest Path: 3 S, A, C, G		The Shortest Path: 3 S, A, C, G	
C	S, A, B, D, G	C, D	S, A, B, D, G	B, D	S, A, C, G	D, B, D	S, A, C, G

UCS, Graph		UCS, Tree	
S		S_0	
A_2, B_3, D_5	S_0	A_2, B_3, D_5	S_0
B_3, D_5, C_6	S_0, A_2	B_3, D_5, C_6	S_0, A_2
D_5, C_6, ~~D_7~~	S_0, A_2, B_3	D_5, C_6, D_7	S_0, A_2, B_3
C_6, G_{10}	S_0, A_2, B_3, D_5	C_6, $D_{7,\,1}$, G_{10}	S_0, A_2, B_3, D_5
G_8, ~~G_{10}~~	S_0, A_2, B_3, D_5, C_6	D_7, G_8, G_{10}	S_0, A_2, B_3, D_5, C_6
Finish	S_0, A_2, B_3, D_5, C_6, G_8	G_8, G_{10}, G_{12}	S_0, A_2, B_3, D_5, C_6, D_7
		G_{10}, G_{12}	S_0, A_2, B_3, D_5, C_6, D_7, G_8
The Shortest Path: S_0, A_2, C_6, G_8		The Shortest Path: S_0, A_2, C_6, G_8	

ج) تفاوت پیچیدگی زمانی BFS برای آزمون‌های هدف متفاوت:

اگر فرض شود هدف در لایه d+1 وجود دارد؛ اگر آزمون هدف،هنگام تولید نود باشد با پیچیدگی زمانی $O(b^d)$ الگوریتم اجرا می‌شود اما اگر آزمون هدف هنگام بسط نود باشد با پیچیدگی زمانی $O(b^{d+1})$ الگوریتم اجرا می‌شود. مدت زمان اجرا بیش‌تر می‌شود اما از آن‌جا که در پیچیدگی زمانی ضریب ثابت اثری ندارد پیچیدگی زمانی در واقع همان $O(b^d)$ باقی خواهد ماند.