

پاسخ تمرین تئوری ۱ درس هوش مصنوعی

امیر حسین عاصم یوسفی
۹۶۱۱۰۳۲۳

۱۳۹۸ اردیبهشت ۲۷

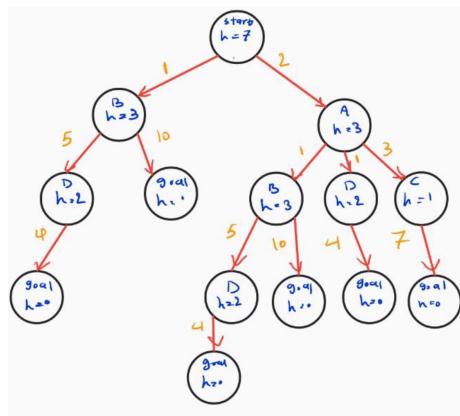
سوال ۱

جدول به صورت زیر می باشد :

	بازیکن شطرينج timed	air hockey ^۵	پازل crossword	حل مکعب روبیک ^۴	Risk ^۳	بازی ساخت پل ^۲	بازی ساخت پل ^۲	بررسیگر ^۱ 20q ^۱
Observable?	Full	Partially	Full	Full	Partially	Partially	Partially	Full
Deterministic?	Yes	No	Yes	Yes	No	Yes	Yes	No
Episodic?	No	No	No	No	No	No	No	No
Static?	Yes(semi)	No	Yes	Yes(semi)	No	Yes(semi)	Yes(semi)	Yes(semi)
Discrete?	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes
Single-Agent?	No	No	Yes	Yes	No	Yes	Yes	Yes

سوال ۲

ابتدا گراف داده شده را به شکل زیر بازنویسی می کنیم :



BFS

با توجه به این که این روش یک روش جستجو Uninformed می باشد بنابراین نیازی به مقدار تابع h و طول بال ها برای انجام این الگوریتم نیست و فقط باید این را درنظر داشته باشیم که باید ترتیب الفبایی را رعایت کنیم . برای اجرای این الگوریتم دو صفت به نام های Expanded و Visited در نظر می گیریم :

مرحله ۱

Visited : S .۱

Expanded : empty .۲

مرحله ۲

Visited : A , B .۱

Expanded : S .۲

مرحله ۳

Visited : B , C , D .۱

Expanded : S , A .۲

مرحله ۴

Visited : C , D , Goal .۱

Expanded : S , A , B .۲

مرحله ۵

Visited : D , Goal .۱

Expanded : S , A , B , C .۲

مرحله ۶

Visited : Goal .۱

Expanded : S , A , B , C , D .۲

مرحله ۷

Visited : empty .۱

Expanded : S , A , B , C , D , Goal .۲

DFS

در این الگوریتم در عمق پایین می رویم تا به گره هدف برسیم و تنها باید ترتیب الفبایی را رعایت کنیم :

مرحله ۱

Visited : S .۱

Expanded : empty .۲

مرحله ۲

Visited : A , B .۱

Expanded : S .۲

مرحله ۳

Visited : C .۱

Expanded : S , A .۲

مرحله ۴

Visited : Goal .۱

Expanded : S , A , C .۲

مرحله ۵

Visited : empty .۱

Expanded : S , A , C , Goal .۲

UCS

در این الگوریتم گره ای را Expand می کنیم Path Cost کمتری دارد :

مرحله ۱

Iteration 1 : [S -> B , 1] , [S -> A , 2]

مرحله ۲

Iteration 2 : [S -> A -> B , 3] , [S -> A -> D , 3] , [S -> A -> C , 5] , [S -> B -> D , 6] , [S -> B -> G , 11]

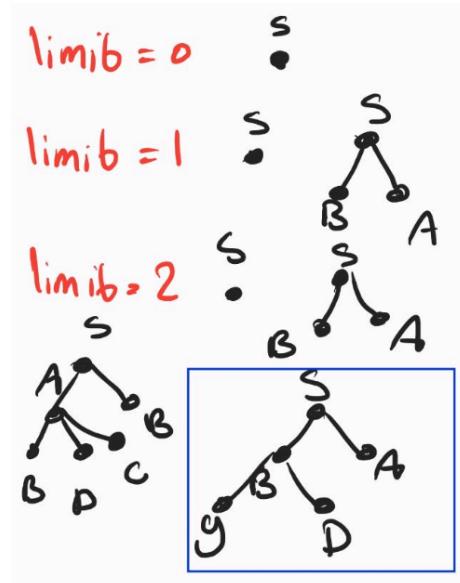
مرحله ۳

Iteration 3 : [S -> A -> B -> G , 13] , [S -> A -> D -> G , 7] , [S -> A -> C -> G -> , 12] , [A -> B -> D -> G , 12] , [S -> B -> G - , 11]

در هر مرحله چون همواره یک لیست مرتب شده داریم بنابراین [S -> A -> D -> G , 7] در اولویت قرار می گیرد بنابراین در مرحله بعدی جواب بهینه که مسیر گفته شده را طی می کند از صفحه بیرون آمده و جستجو تمام می شود .

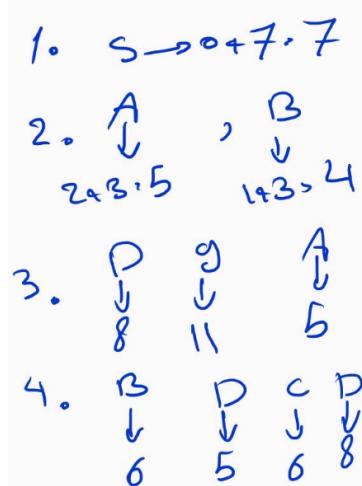
ID DFS

برای این الگوریتم باید هر بار با یک Limit جدید به جستجو به صورت DFS ادامه دهیم . که به صورت زیر است :

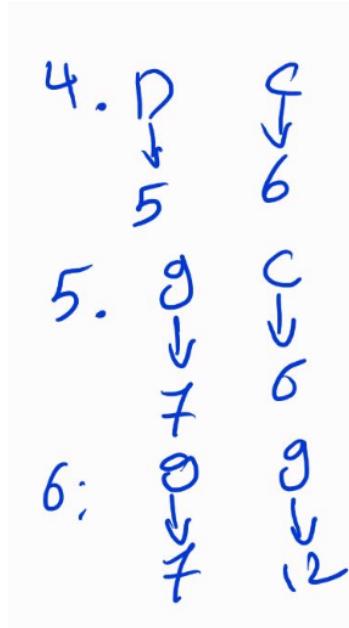


A *

برای این الگوریتم هر بار باید مقدار $F = g + h$ را برای هر گره حساب کنیم و هر بار گره ای را Expand می کنیم که کمترین F را دارد .
در این الگوریتم اگر به یک حالت تکراری با مقدار F بهتر رسیدیم که هنوز Expand نشده است مقدار بهتر را جایگزین می کنیم .



در این مرحله چون یک بار گره B را Expand کرده ایم آن را از صف حذف می کنیم . همچنان در این حالت همان طور که می توان دید به گره D با دو مقدار مختلف رسیدیم که یکی از طریق گره A با مقدار $F = 5$ و دیگری از مسیر گره B با مقدار $F = 8$ که در اینجا بهترین گره رو نگه می داریم و مراحل به صورت زیر می شود :



و در مرحله ۶ ام با دو مقدار متفاوت F به گره هدف رسیدیم که در اینجا بهترین گره را نگه میداریم بنابراین Expansion Order به صورت زیر می باشد :

$$S, F = 7 \text{ .1}$$

$$B, F = 4 \text{ .2}$$

$$A, F = 5 \text{ .3}$$

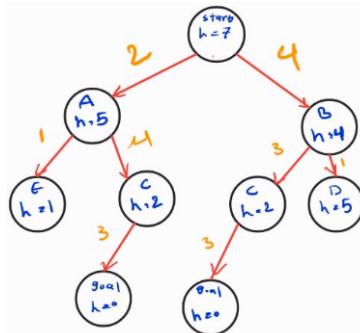
$$D, F = 5 \text{ .4}$$

$$C, F = 6 \text{ .5}$$

$$\text{Goal}, F = 7 \text{ .6}$$

سوال ۳

ابتدا گراف داده شده را به صورت زیر بازنویسی می کنیم :



Best - First Search

در این روش تنها به مقدار Heuristic توجه می کنیم و همواره گره ای را Expand می کنیم که کمترین Heuristic را دارد . که به صورت زیر می باشد :

$$S, h = 7 \text{ .} 1$$

$$B, h = 4 \text{ .} 2$$

$$C, h = 2 \text{ .} 3$$

$$G = 0 \text{ .} 4$$

بنابراین مسیری که این الگوریتم پیدا می کند به صورت $S \rightarrow B \rightarrow C \rightarrow G$ می باشد .

A *

برای این الگوریتم Expansion Order به صورت زیر می باشد :

$$S, F = 7 \text{ .} 1$$

$$A, F = 7 \text{ .} 2$$

$$E, F = 4 \text{ .} 3$$

$$B, F = 8 \text{ .} 4$$

$$C, F = 8 \text{ .} 5$$

$$\text{Goal}, F = 9 \text{ .} 6$$

بنابراین مسیری که این الگوریتم طی میکند به صورت $S \rightarrow A \rightarrow C \rightarrow G$ می باشد

IDA *

در این الگوریتم هر بار یک Limit جدید برای مقدار F مشخص می کنیم اگر در آن مرحله به گره هدف رسیدیم که کار تمام است ولی اگر این اتفاق نیافتد از بین آن هایی که از محدودیت ما بیشتر هستند کمترین را برای Limit مرحله بعد انتخاب می کنیم بنابراین داریم :

Limit = 7

S , F = 7 .۱

A , F = 7 .۲

E , F = 4 .۳

در این مرحله به گره هدف نرسیدیم پس این بار Limit = 8

Limit = 8

S , F = 7 .۱

A , F = 7 .۲

E , F = 4 .۳

C , F = 8 .۴

در این مرحله فرزند C که گره هدف است مقدار F = 9 را دارد بنابراین الگوریتم با توجه به محدودیت آن را از صف بیرون نمی آورد پس این بار با Limit = 9 اجام می دهیم.

Limit = 9

S , F = 7 .۱

A , F = 7 .۲

E , F = 4 .۳

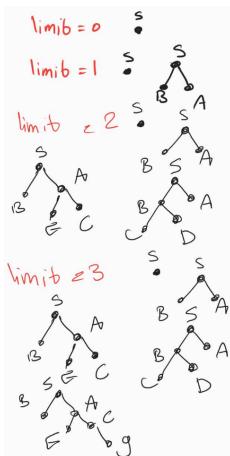
C , F = 8 .۴

G , F = 9 .۵

بنابراین مسیری که این الگوریتم طی میکند به صورت S -> A -> C -> G می باشد

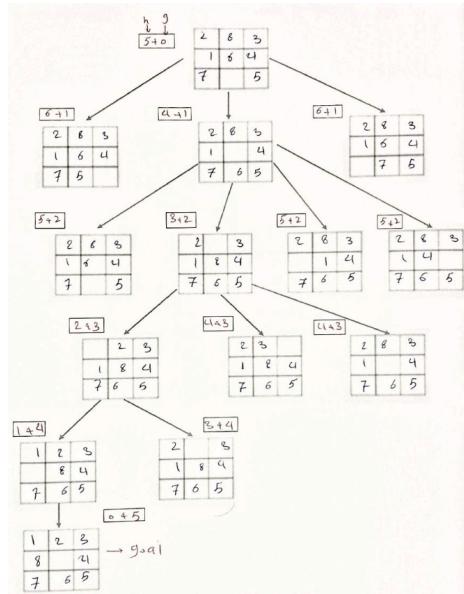
ID DFS

این الگوریتم به صورت زیر می باشد :



سوال ۴

الف

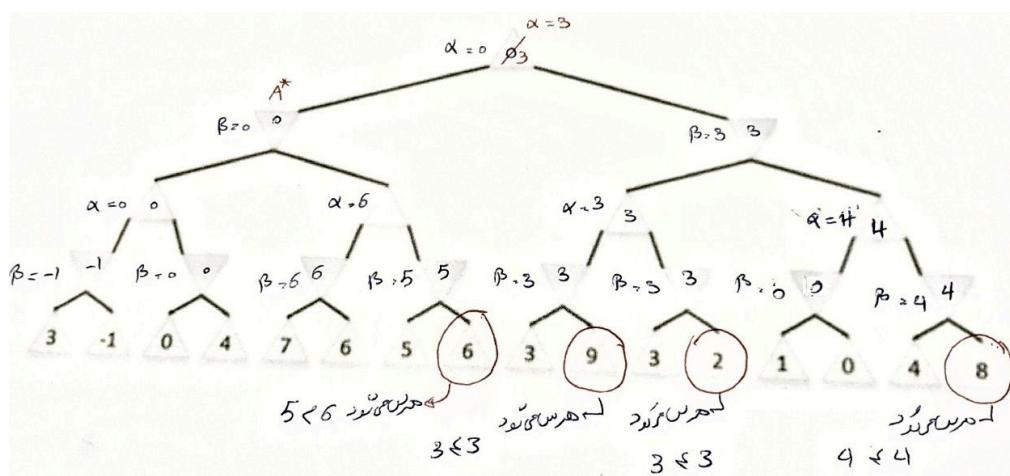


ب

خیر، زیرا ممکن است در Local Maxima به دام بیافتیم و هیچ گاه به بلندترین قله نرسیم
دلیل دیگر این است که ممکن است فلات داشته باشیم یعنی مقدار Heuristic تغییری نکند.

سوال ۵

با اجرای الگوریتم گفته شده به نتیجه زیر می رسمیم :



سوال ۶

الف

با توجه به فرضی که در قسمت ب گفته شده اگر این مسئله را به روش ارضا محدودیت حل کنیم نیازی به Backtrack نداریم زیرا در هر مرحله با توجه به این ترتیب الفبایی هم در مقدار دادن به کشورها و هم در رنگ ها وجود همواره در اولین مرحله به افغانستان (ترتیب الفبایی در کشورها) رنگ آبی (ترتیب الفبایی در رنگ ها) و بنابراین مقدار دهی به ترتیب زیر می باشد :

Afghanistan .۱

Iran .۲

Iraq .۳

Turkey .۴

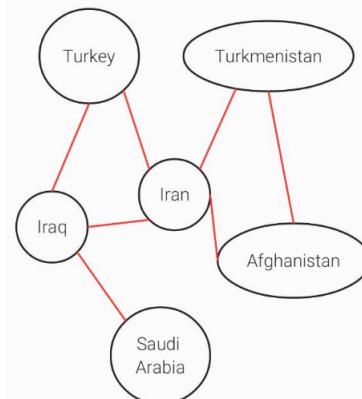
Turkmenistan .۵

Saudi Arabia .۶

اما اگر فرض موجود در قسمت ب وجود نداشت می توانستیم ابتدا به افغانستان و بعد به ترکمنستان و بعد ترکیه مقدار دهیم آنگاه هیچ مقداری برای ایران باقی نمی ماند و در این صورت مجبور به Backtrack بودیم .

ب

برای این قسمت ابتدا گراف محدودیت را رسم می کنیم که به صورت زیر می باشد :



حال با توجه به این گراف محدودیت روش های گفته شده را پیاده سازی می کنیم .

FC

Step	Afghanistan	Iran	Iraq	Turkey	Turkmenistan	Saudi Arabia
۰	R, G, B	R, G, B	R, G, B	R, G, B	R, G, B	R, G, B
۱	B	R, G	R, G, B	R, G, B	R, G	R, G, B
۲	B	G	R, B	R, B	R	R, G, B
۳	B	G	B	R	R	R, G

PL

در این الگوریتم در هر مرحله فقط تاثیر دو متغیر را بر یکدیگر بررسی می کنیم . همچنین در این الگوریتم همواره بعد از مقداردهی Forward Checking انجام می دهیم . در مرحله اول به افغانستان مقدار می دهیم و داریم :

Step	Afghanistan	Iran	Iraq	Turkey	Turkmenistan	Saudi Arabia
0	R, G, B	R G B	R G B	R G B	R G B	R G B
1	B	R G	R G B	R G B	R G	R G B

در همین مرحله تاثیر دو متغیری که مقدار نگرفته اند (ایران و عراق) را برهم بررسی می کنیم و داریم :

1	B	R G PL	R G B PL	R G B	R G	R G B
---	---	-----------	-------------	-------	-----	-------

در مرحله دوم با توجه به تصویر بالا رنگ قرمز را به ایران می دهیم و حال با استفاده از Forward Checking داریم :

2	B	B G PL	R G B FC PL	R G B	R G B	R G B
---	---	-----------	----------------	-------	-------	-------

که در نهایت برای مرحله دو به جدول زیر می رسیم :

2	B	R	B	G B FC	G	R G B FC
---	---	---	---	--------------	---	-------------

در مرحله سوم حال به عراق تنها مقدار باقی مانده برای آن (آبی) را اختصاص می دهیم و با استفاده از Forward Checking داریم :

3	B	R	B	G	G	R G
---	---	---	---	---	---	-----

FL

در این الگوریتم تاثیر هر دو متغیری که مقدار نگرفته اند را نسبت به هم بررسی می کنیم . و در هر مرحله بعد از این که از دامنه متغیرها با استفاده از Fully Look Ahead مقداری غیر مجاز را حذف کردیم و بعد از مقدار دادن به یک متغیر الگوریتم Forward Checking را اعمال کنیم . که با انجام دادن این مراحل به جدول زیر می رسیم :

Step	Afghanistan	Iran	Iraq	Turkey	Turkmenistan	Saudi Arabia
0	R G B	R G B	R G B	R G B	R G B	R G B
1	B	R O FL	R O B	R O B	R O FL	R G B
2	B	R	B	R	R	R O B
3	B	R	B	R	R	R G