

بنام خدا

مریم رضوانی 9921160019

تمرین سری سوم

1) این واژه ها را تعریف کنید.

حالت :

حالت از طریق مکان های موجود در مسئله که عامل می تواند در آن قرار بگیرد، تعریف می شود.

فضای حالت:

مجموعه ای از تمام حالت هایی است که با اجرای دنباله ای از فعالیت ها ، از حالت شروع قابل دستیابی هستند.

درخت جستجو:

این درخت با استفاده از حالت ابتدایی و تابع پسین که همراه یکدیگر فضای حالت را تعریف می کنند، ایجاد می شود.

گره جستجو:

نشان دهنده ی حالت موجود در فضای حالت مسئله است.

ریشه درخت جستجو، گره جستجویی است که با حالت ابتدایی متناظر است.

هدف:

هدف را مجموعه ای از حالت های دنیا در نظر می گیریم (آن حالت هایی که هدف در آنها برآورده می شود). وظیفه عامل یافتن دنباله ای از فعالیت ها است که آن را به "حالت هدف" می رساند.

فعالیت:

کارهای یا اقدام هایی که عامل انجام می دهد و ممکن است منجر به تغییر حالت شود یا نشود(یال ها)

تابع پسین:

یک حالت را بعنوان ورودی می گیرد و سپس مجموعه ای از زوج مرتب های (حالت،اقدام) تولید می کند.که نشان می دهد به ازای هر حالت چه فعالیتی باید عامل انجام دهد.

ضریب انشعاب:

b، حداکثر تعداد گره های پسین یک گره است. تعداد فعالیت هایی که عامل می تواند در هر حالت انتخاب کرده و انجام دهد.

(2) توضیح دهید چرا تدوین مسئله باید پس از تدوین هدف بیاید؟

اینکار باعث می شود تا هنگام اجرای حالت ها هنگام تدوین مسئله بین اینکه به کدام حالت بعدی باید برویم تا به حالت نهایی برسیم هدفی (بدانیم چه حالتی اجتناب یا انتخاب کنیم) داشته باشیم و آن را دنبال کنیم در صورتی که اگر چنین نبود بین اینکه به کدام حالت برویم دچار سردرگمی می شدیم.

(3) فرض کنید که (LEGAL_ACTION) مجموعه اقداماتی که در حالت s مجاز هستند و RESULT(a,s) حالتی که از انجام اقدام مجاز a در حالت s نتیجه می شود را نشان دهند.

SUCCESSOR-F را برحسب LEGAL_ACTIONS و result و برعکس تعریف کنید.

```
# SUCCESSOR-FN() defined in terms of result and LEGAL_ACTIONS
def SUCCESSOR-FN(s):
    return [(a,result(a,s)) for a in LEGAL_ACTIONS(s)]
# LEGAL_ACTIONS() and result defined in terms of SUCCESSOR-FN()
def LEGAL_ACTIONS(s):
    return [a for (a,s) in SUCCESSOR-FN(s)]
def result(a,s):
    for (a1,s1) in SUCCESSOR-FN(s):
        if(a==a1):
            return s1
```

(4) نشان دهید حالت های معماری پازل هشت به دو مجموعه ی جدا از هم تقسیم می شوند. بطوریکه در یک مجموعه می توان از هر حالت به حالت دیگر رسید. ولی از هیچ حالتی در مجموعه نمیتوان به هیچ مجموعه دیگر رسید. رویه ای ارائه دهید که مشخص کند یک حالت مورد نظر در کدام مجموعه قرار دارد و توضیح دهید که چرا این کار برای تولید حالت های تصادفی مفید است؟

	1	2
3	4	5
6	7	8

حالت هدف را بصورت خانه های مرتب شده روبرو در نظر می گیریم. برای بررسی مرتب بودن از گوشه سمت چپ سطر اول شروع می کنیم و بترتیب خانه هارا بررسی می کنیم و به سطر بعد میرویم الی آخر اگر دو مقدار دو خانه مربع ها نامرتب بود بایکدیگر تعویض می شوند و به همین ترتیب.

فرض می کنیم n نشان دهنده مجموع کل تعویض ها بعلاوه شماره خانه سطر خالی باشد؛ (طبق شکل که خانه ها مرتب اند و تعویضی نیاز نداریم n=1). بنابراین دو نوع چیدمان نیاز داشت چیدمان

هایی که فرد و چیدمان های که زوج هستند چون اضافه کردن عددی زوج تغییری در مقدار زوج یا فردی اعداد این چیدمان ها ایجاد نخواهد کرد. (برای حالتی که n فرد است و یا n زوج است).

راه حلی که برای این مسئله وجود دارد این است که از زوج یا فرد بودن n مطمئن شویم چون یافته اند که تغییر در مجموع n همواره زوج است. و هر عددی که با عدد زوج جمع شود تغییری در زوج یا فرد بودن آن ایجاد نمی شود. بنابراین قبل از حل پازل باید مقدار n را برای حالت هدف و حالت شروع تعیین کنیم. و از نظر زوج یا فرد بودن آنها را جدا کنیم تا مشکلی پیش نیاید در غیر اینصورت مسئله قابل حل نخواهد بود.

5) مسئله n وزیر را با استفاده از فرموله کردن افزایشی کارآمد که در بخش 1-2-3 آمده است در نظر بگیرید توضیح دهید چرا فضای حالت حداقل $\sqrt[3]{n!}$ حالت دارد و بزرگترین مقدار n را برآورد کنید. که جستجوی جامع امکان پذیر باشد. (راهنمایی: با در نظر گرفتن حداکثر تعداد مربع هایی که وزیر می تواند در هر ستون گارد دهد یک کران پایین برای ضریب انشعاب پیدا کنید).

هر وزیر را در ستونی مجزا از سمت چپ قرار می دهیم به این ترتیب $n!$ حالت وزیر قرار دارد. چون در تدوین این مسئله هر وزیر جدید باید در خانه ای قرار بگیرد که توسط وزیر های خانه دیگر تهدید نشود پس هر وزیر سه خانه از ستون های بعدی در صفحه را تهدید میکند (مورب بالا، مورب پایین، همان سطر بصورت افقی). بنابراین وزیر اول ستون اول، وزیر دوم ستون دوم با $(n-3)$ حالت امن و وزیر سوم بعلت دو وزیر قبلی که تهدید می شود $(n-6)$ حالت می تواند قرار گیرد و... به همین منوال. در نتیجه فضای حالت داریم: $S^3 \geq n(n-3)(n-6) \dots$

$$S^3 \geq n.n.n.(n-3).(n-3).(n-3).(n-6).(n-6).(n-6) \dots$$

$$\geq n.(n-1).(n-2).(n-3).(n-4).(n-5).(n-6).(n-7).(n-8) \dots$$

$$= n!$$

$$S \geq \sqrt[3]{n!}$$

6) آیا همیشه یک فضای حالت متناهی به یک درخت جستجوی متناهی منجر می شود؟ خیر

یک فضای حالت متناهی که به شکل درخت است چطور؟ بله

آیا می توانید بطور دقیق بیان کنید که چه نوع فضاهای حالتی همیشه به درختهای جستجوی متناهی منجر می شوند؟

یک فضای حالت متناهی همیشه ممکن نیست به یک درخت جستجوی متناهی منجر شود. مثال نقشه شهر رومانی در کتاب. که چون گراف جهت ندارد ممکن است تا بی نهایت برای یک حالت بین دو شهر جابجا شویم. و علی رغم فضای حالت متناهی، درختی نامتناهی داشته باشیم

اما یک فضای حالت متناهی که به شکل درخت است چون حلقه بی نهایت تشکیل نمی شود متناهی خواهد بود.

7) برای هریک از موارد زیر، مسئله را بطور کامل فرموله کنید. فرموله کردن طوری دقیق باشد که بتوان آن را پیاده سازی کرد؟

الف) با استفاده از 4 رنگ باید نقشه پیچیده ای را رنگ کنید ، بطوریکه هیچ یک از دو ناحیه همجوار رنگ یکسانی نداشته باشد؟

حالت شروع: هیچ منطقه ای رنگ آمیزی نشده است.

آزمون هدف: تمام مناطق رنگ آمیزی شده باشند و هیچ دومتطقه مجاوری هم رنگ نباشد.

تابع پسین: انتساب یک رنگ به منطقه

تابع هزینه: تعداد انتساب ها

ب) میمونی با سه فوت قد در اتاقی قرار دارد که موزهایی از سقف 8 فوتی آویزان است . میمون می خواهد موزها را بگیرد. داخل اتاق دو جعبه به ارتفاع 3 فوت وجود دارد که قابل جابجایی بوده روی هم قرار می گیرند.

حالت شروع: میمون داخل یک اتاق

آزمون هدف: میمون موز را گرفته باشد

تابع پسین: هل دادن جعبه از یک محل به محل دیگر

تابع هزینه: تعداد فعالیت های انجام شده

پ) برنامه ای دارید که وقتی رکوردی را از فایلی دریافت می کند پیام می دهد، " رکورد ورودی معتبر است " پردازش هر رکورد مستقل از رکورد های دیگر است . می خواهید مشخص کنید که کدام رکورد نامعتبر است؟

حالت شروع: رکورد های ورودی

آزمون هدف: نمایش پیام " رکورد نامعتبر است."

تابع پسین: اول نصفه ابتدایی رکوردها عملیات اجرا را تکرار کن و بر روی نصفه دیگر عملیات اجرا را تکرار کن.

تابع هزینه: تعداد دفعات اجرا

ت) 3 پارچ دارید که اندازه ی آنها 12 گالن ، 8گالن ، و 3 گالن است. شیر آب نیز در دسترس است.می توانید پارچ هارا خالی یا پر کنید .از یکی به یکی دیگر یا بر روی زمین بریزید.باید یک گالن را اندازه گیری کنید (در یکی از پارچ ها فقط یک گالن باقی بماند)

حالت شروع: سه پارچ با مقدار $[0,0,0]$

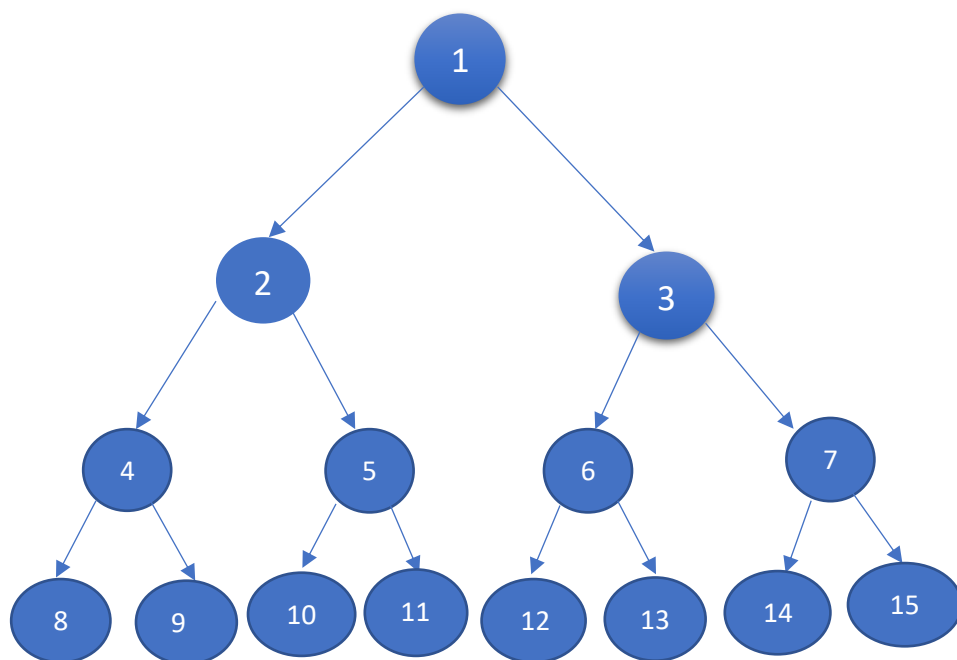
آزمون هدف: اندازه گیری گالن

تابع پسین:به ازای $[x,y,z]$ تولید حالت های $[x,8,z], [x,y,3], [12,y,z]$ با پرکردن هرپارچ، تولید حالت های $[x,y,0], [0,y,z], [x,0,z]$ با خالی کردن هرپارچ به ازای دو پارچ x,y آب درون y را درون پارچ x بریزیم که موجب می شود پارچ x دارای حداقل $x+y$ یا y آب باشد.کاهش حجم کوزه اول توسط کوزه y .

تابع هزینه:تعداد دفعات انجام

8) فضای حالتی در نظر بگیرید که در آن ، حالت شروع شماره 1 است و تابع پسین برای حالت n ، دو حالت به شماره های $2n, 2n+1$ را برمی گرداند.

الف) فضای حالت را برای حالت های 1 تا 15 مشخص کنید.



ب) فرض کنید که حالت هدف، 11 است. ترتیبی که گرهما ملاقات می شوند را برای جستجوی اول سطح، جستجوی عمق محدود با محدودیت 3 و جستجوی عمیق شونده تکراری فهرست کنید.

جستجوی اول سطح: 1 2 3 4 5 6 7 8 9 10 11

جستجوی عمقی محدود: 1 2 4 8 9 5 10 11

جستجوی عمیق شونده تکراری: 1, 123, 1 2 4 5 3 6 7, 1 2 4 8 9 5 10 11

ج) آیا جستجوی دو طرفه برای این مسئله مناسب است؟ اگر چنین است به تفصیل توضیح دهید که چگونه عمل می کند؟

بله جستجوی دو طرفه خوب است چون تنها پسین حالت n در جهت معکوس حالت $[n/2]$ است که به جستجوی سریعتر کمک می کند.

د) ضریب انشعاب در هر جهت جستجوی دو طرفه چیست؟

در جهت مستقیم ضریب 2 و در جهت معکوس ضریب 1

ه) آیا پاسخ (ج) یک تدوین مجدد برای مسئله را پیشنهاد می کند که اجازه می دهد مسئله رسیدن از حالت 1 به یک هدف خاص، تقریباً بدون هیچ جستجویی حل شود؟

بله از حالت شروع؛ شروع می کنیم و واکنش پسین معکوس را تاجایی ادامه می دهیم که به حالت 1 دست پیدا کنیم.

9) مسئله کشیش و آدمخوار به این صورت است: سه کشیش و سه آدمخوار در یک طرف رودخانه قرار دارند. قایقی وجود دارد که هر بار یک یا دو نفر می توانند بر آن سوار شوند. و به طرف دیگر بروند بدون اینکه در هر طرف تعداد کشیشان کمتر از تعداد آدمخواران باشد. این مسئله در AI به این علت معروف است که موضوع اولین مقاله ای بود فرموله کردن مسئله را از دیدگاه تحلیلی بررسی کرد.

الف) مسئله را دقیقاً فرموله کنید. فقط آن بخش هایی را مشخص کنید که به جواب معتبری منجر می شود. نمودار کامل فضای حالت را بررسی کنید.

فضای حالت: یک دنباله مرتب شده از اعداد است که تعداد کشیش ها، تعداد آدمخوار ها و محل قایق از رودخانه از آنجا که مسئله شروع شده را نمایش می دهد.

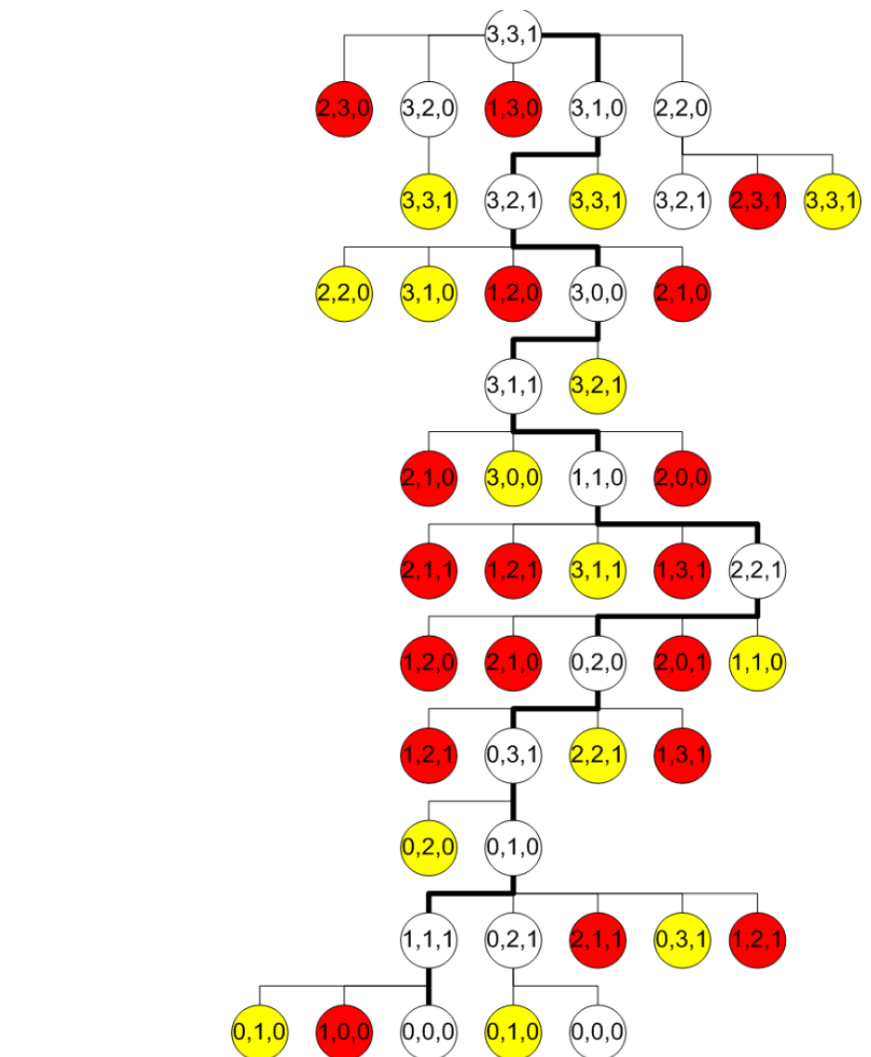
که بصورت یک مجموعه سه تایی از اعداد (X, Y, Z) است که X تعداد آدم ها در طرف رودخانه و Y تعداد آدمخوارها در طرف رودخانه و Z اگر یک باشد یعنی قایق طرف رودخانه و اگر صفر باشد طرف دیگر رودخانه را نشان می دهد.

تابع جانشین: از هر حالت، عملگرهای ممکن یک کشیش و یک آدمخوار، دو کشیش و دو آدمخوار، یا یکی از هرکدام را درقایق قرار می دهند.

آزمون هدف: رسیدن به حالت $(0,0,0)$

تابع هزینه: تعداد دفعات عبور از رودخانه

$(3,3,1) \rightarrow (2,2,0) \rightarrow (3,2,1) \rightarrow (3,0,0) \rightarrow (3,1,1) \rightarrow (1,1,0) \rightarrow (2,2,1) \rightarrow (0,2,0) \rightarrow (0,3,1) \rightarrow (0,1,0) \rightarrow (0,2,1) \rightarrow (0,0,0)$



ب) مسئله را با استفاده از یک الگوریتم جستجوی مناسب بطور بهینه پیاده سازی و حل کنید. آیا بررسی حالت های تکراری ایده ی جالبی است؟

چون فضای حالت کوچک است هر الگوریتمی می تواند بهینه باشد. حالت های تکراری که منجر به بازگشت به حالت های قبلی دیده شده می شوند باید حذف کرد.

پ) با توجه به اینکه فضای حالت کوچک است چرا فکر می کنید حل این معما دشوار است؟
بخطرات اینکه تشخیص حرکت ها قطعی و مطمئن نیستیم ممکن است در حرکتی آدم ها توسط آدمخوار ها خورده شوند یا در حرکتی به حالت قبلی برگردیم. و این به علت بزرگی فاکتور انشعاب در این مسئله است که راهی برای پیشروی در آن نداریم.

10) سوال پیاده سازی است پس حذف است.

دو نسخه از تابع پسین برای پازل 8 تایی پیاده سازی کنید: یکی تمامی پسین ها را در یک لحظه با کپی کردن و ویرایش ساختار داده پازل 8 تایی تولید می کند و دیگری در هربار فراخوانی یک پسین جدید می سازد و با تغییر مستقیم حالت والد کار می کند. نسخه هایی از جستجوی اول عمق عمیق شونده تکراری بنویسید که از این توابع استفاده می کنند. و کارایی آنها را مقایسه کنید.

11) در صفحه 95، درباره جستجوی طولانی کننده تکراری صحبت کردیم. (الگوریتمی تکراری که مشابه جستجو هزینه یکنواخت است؟) این روش براساس استفاده از محدودیتهای در حال افزایش در مورد هزینه ها است. اگر گرهی تولید شود که هزینه مسیری از محدودیتهای فعلی بیشتر شود بلافاصله کنار گذاشته می شود. در هر تکرار جدید این محدودیت براساس کمترین هزینه مسیر در تمامی گره هایی که در تکرار قبل کنار گذاشته اند تعیین می شود.

الف) نشان دهید این الگوریتم برای حالت کلی هزینه مسیر عمومی بهینه است؟

جستجوی عمیق شونده تکراری تضمین بهینگی را از الگوریتم جستجوی هزینه یکنواخت به ارث می برد. ولی به آن اندازه به حافظه احتیاج ندارد. اساس کار بدین صورت است که به جای محدودیتهای عمق در حال افزایش، از محدودیتهای هزینه مسیر در حال افزایش استفاده کنیم.

در این الگوریتم بسط گره ها بترتیب افزایش هزینه ها صورت می گیرد بنابراین اولین هدف گره ای با کمترین هزینه خواهد بود و بهینه است.

ب) یک درخت را با ضریب انشعاب b عمق راه حل d و هزینه گام واحد در نظر بگیرید.

روش طولانی کننده تکراری به چند تکرار نیاز دارد؟

به d تکرار نیاز دارد و از مرتبه $O(b^d)$ می باشد.

ج) اکنون هزینه گامهایی را در نظر بگیرید که از بازه پیوسته $(0,1)$ با حداقل هزینه مثبت ممکن ϵ گرفته می شوند. در بدترین حالت به چند تکرار نیاز است؟

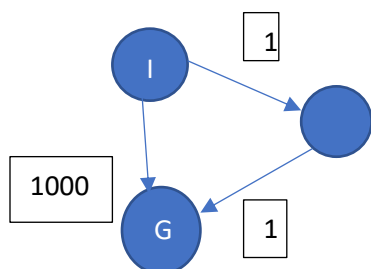
d/ϵ

د) پیاده سازی است پس سوال حذفی است.

12) ثابت کنید که جستجوی هزینه یکنواخت و جستجوی اول سطح با هزینه گام ثابت، وقتی که با الگوریتم GRAPH – SEARCH استفاده می شوند، بهینه هستند. یک فضای حالت با هزینه گام ثابت را نشان دهید که در آن GRAPH – SEARCH با استفاده از روش عمیق شونده تکراری، یک راه حل نیمه بهینه را پیدا می کند.

جستجوی هزینه یکنواخت با جستجوی عمق اول برابر است در صورتیکه همه هزینه های مسیر برابر باشند، شما فقط می توانید یکی از این دو روش را ثابت کنید. از آنجایی که هزینه های هر مرحله ثابت است، وقتی یک گره بازدید نشده با جستجوی عرضی اول (جستجوی هزینه یکنواخت) بازدید می شود، هزینه کمترین هزینه خواهد بود. علاوه بر این، از آنجایی که ما از الگوریتم GRAPH-SEARCH استفاده می کنیم، هیچ گره ای بیش از یک بار بازدید نخواهد شد. بنابراین اگر گره هدف بازدید شود، راه حل بهینه را دریافت خواهیم کرد. فضای حالتی که GRAPH-SEARCH با روش جستجوی عمیق شونده تکراری حل می شود الگوریتم در یافتن راه حل بهینه شکست خواهد خورد در شکل زیر نشان داده شده است که در آن حالت اولیه و G حالت هدف است:

اگر مسیری با یک گام از A به G داشته باشیم با هزینه 1000 ولی مسیر دیگری با گذر از دو حالت و هریک هزینه 1 است که به هدف می رسد می توانیم نشان دهیم که راه حلی نیمه بهینه با هزینه گام متغیر داریم.



13) فضای حالتی را توصیف کنید که در آن، جستجوی عمیق شونده تکراری، بسیار بدتر از

جستجوی اول عمق عمل می کند. (مثلاً $O(n^2)$ در مقابل $O(n)$).

اگر فضای حالت شما مانند یک لیست پیوندی است (یعنی فقط 1 فرزند در هر گره) و حالت هدف برگ

است، دقیقاً به موقعیتی خواهید رسید که توضیح دادید.

فضایی را در نظر بگیرید که یک هدف در عمق n داریم و هر حالت یک حالت بعدی داشته و در اینصورت با جستجوی اول سطح با $O(n)$ به جواب میرسیم در حالیکه در جستجوی اول عمق در اولین تکرار فقط فرزند ریشه را ملاقات خواهید کرد در تکرار دوم، فرزند ریشه و فرزند خود را مشاهده خواهید کرد (عمق = 2، بازدید از 2 گره). در تکرار سوم، با بازدید از 3 گره، به عمق 3 می روید. از این رو تعداد کل بازدیدها $1+2+3+...+n=O(n^2)$

14) برنامه ای بنویسید که دو URL صفحه وب را بعنوان ورودی می گیرد و یک مسیر از پیوندها از یکی به دیگری می یابد. چه راهبرد جستجویی مناسب است؟ آیا جستجوی دوطرفه ایده خوبی است؟ آیا می توان از یک موتور جستجو برای پیاده سازی یک تابع پیشین استفاده کرد؟

یک آدم معمولی یا یک عامل فقط زمانی می تواند بفهمد صفحه بعد یک لینک چیست که به آن مراجعه کند. پس خوب است از جستجوی اول سطح یا روش حریصانه مانند اول بهترین استفاده شود تا بر حسب تعداد لغات بین صفحات ابتدایی و صفحه هدف عملیات جستجو را انجام دهد. تا شاید در زمینه رسیدن لینک ها به صفحه هدف کار ساز باشد.

در مورد موتور های جستجو میدانیم که گراف کاملی از صفحه وب را در اختیار دارند بطوریکه میتوانند کاربر را به تمام صفحات یا بخشی از صفحاتی که به یک صفحه لینک شده اند هدایت کنند پس مشکل انسان ها را نداشته و استفاده از جستجوی دوطرفه در آنها ایده مناسبی است.

تکنیک جستجوی دوطرفه یک گراف جهت دار کوتاه ترین مسیر را بین دو مکان راس شروع تعیین می کند. دو جستجوی همزمان انجام می دهد، یکی از نقطه شروع به جلو و دیگری از مقصد به سمت عقب حرکت می کند و زمانی که همگرا می شوند متوقف می شود.

هدف از جستجوی دو طرفه تعیین کوتاه ترین مسیر بین راس شروع ثابت و راس پایانی با انجام همزمان دو پیمایش نمودار (BFS) است. این روش سریع تر است و زمان لازم برای پیمایش نمودار را کاهش می دهد. کاربردهای دیگری نیز برای آن وجود دارد.

15) مسئله یافتن کوتاهترین مسیر را بین دو نقطه در صفحه نظر بگیرید که دارای موانعی به شکل چند جمله ای محدب مانند شکل 31-3 است. این وضعیت حالت ایده آل برای مسئله ای است که روبات باید حل کند تا در یک محیط شلوغ حرکت کند. (شکل کتاب)

الف) فرض کنید فضای حالت شامل تمام موقعیت های (x,y) در صفحه است چند حالت وجود دارد؟ چند مسیر به هدف وجود دارد؟

اگر تمام نقاط (x,y) به عنوان فضای حالت در نظر گرفته شود آن گاه تعداد حالات و تعداد مسیرها بی نهایت خواهد بود.

ب) توضیح دهید که چرا کوتاهترین مسیر از یک راس چند ضلعی به هر راس دیگر در صحنه، باید شامل خطوط مستقیمی باشد که بعضی از راس های چند ضلعی را بهم متصل می کند. اکنون یک فضای حالت خوب تعریف کنید. اندازه این فضای حالت چقدر است؟

همانطور که می دانید کوتاهترین فاصله بین دو نقطه خط مستقیمی است که از آن دو می گذرد.

فضای حالت برای خطوط مستقیم در این مسئله به دلیل وجود رئوس مختلف چند ضلعی ها دارای مانع است در نتیجه 32 مانع داریم که نقاط تماس باید رئوس مسیر باشد که کل مسیر رأس به رأس ادامه یابد و فضای حالت برای چنین محیطی مختصات رئوس است.

پ) توابع لازم را برای پیاده سازی مسئله جستجو تعریف کنید. یک تابع پسین پیاده سازی کنید که راسی را بعنوان ورودی دریافت کرده مجموعه ای از راس هارا برمیگرداند که طریق خط مستقیم، از یک راس مورد نظر قابل دستیابی است. همسایه هارا در یک چند ضلعی در نظر داشته باشید. برای تابع ابتکاری از فاصله ی خط مستقیم استفاده کنید.

تابع جانشین حالات مختلف در نظر می گیرد و حالت هدف می رساند و اگر در گره نود (شروع) قرار داشته باشیم برای رئوسی که در طی یک واکنش باهم ارتباط دارند داریم:

گره شروع = S

گره هدف = G

تعداد هزینه های رئوس + هزینه مسیر نود = هزینه مسیر

ت) یک یا چند الگوریتم این فصل را برای حل این نوع مسئله ها بکار ببرید. و کارایی آنها را شرح دهید.

معیار	دو طرفه	عمیق شونده تکراری	عمق محدود	اول عمق	هزینه یکنواخت	اول سطح
کامل؟	بله	بله	خیر	خیر	بله	بله
زمان	$O(b^{d/2})$	$O(b^d)$	$O(b^l)$	$O(b^m)$	$O(b^{[c*/c]})$	$O(b^{d+1})$
فضا	$O(b^{d/2})$	$O(bd)$	$O(bl)$	$O(bm)$	$O(b^{[c*/c]})$	$O(b^{d+1})$
بهینه؟	بله	بله	خیر	خیر	بله	بله

16) می توانیم مسئله هدایت در تمرین 3-15 را به محیطی به شکل زیر تغییر دهیم:

- یافته ها فهرستی از موقعیتهای راسهای قابل رویت نسبت به کارگزار خواهد بود. یافته ها موقعیت روبات را شامل نمی شود. روبات موقعیت خودش را باید از نقشه بفهمد فعلا می توان فرض کرد که هر محل چشم انداز متفاوتی دارد.
- هر اقدام برداری خواهد بود که یک مسیر مستقیم برای دنبال کردن را توصیف می کند اگر این مسیر مانعی سرراش نباشد این اقدام موفق خواهد بود. در غیر اینصورت روبات در اولین نقطه ای که مسیرش به یک مانع برخورد کند متوقف می شود. اگر کارگزار یک بردار حرکت صفر برگرداند و در هدف باشد (که ثابت و معلوم است) آنگاه محیط باید اجبارا کارگزار را به یک محل تصادفی (که داخل یک مانع نباشد) منتقل کند.
- مقیاس کارایی برای هر واحد مسافت که طی شود 1 نمره از کارگزار کم می کند و هر بار که به هدف برسد 1000 نمره جایزه می دهد.

الف) پیاده سازی است پس حذفی است.

ب) کارایی کارگزار خود را ثبت کنید (باوادر کردن کارگزار به تولید یادداشتهای مناسب، در حین حرکت) و کارایی آن را در بیش از 100 مرحله ثبت کنید.

کارایی کارگزار توسط حالات باورش تعیین می شود اگر حالت فیزیکی داشته باشیم آنگاه 2s حالت عقیده داریم کارایی کارگزار به ازای هر واحد مسافت پیموده شده از 100 واحد یک امتیاز به معیار کارایی اش اضافه می شود و در انتهای 100 مرحله 100 امتیاز به ازای هر مرحله یک امتیاز به معیار کارایی اش اضافه می شود.

ج) محیط را به نحوی تغییر دهید که 30 درصد مواقع کارگزار در یک مقصد ناخواسته کارش پایان یابد (که بطور تصادفی از راسهای قابل رویت دیگر در صورت وجود انتخاب می شود و در غیر اینصورت اصلاً حرکت نمی کند) مدل فوق یک مدل خاص از خطاهای حرکتی یک روبات واقعی است. کارگزار را به نحوی تغییر دهید که اگر چنین خطایی شناسایی شد متوجه شود که کجاست و آنگاه طرحی بریزد برای اینکه به آنجایی که بوده است برگردد و طرح قبلی را از سر بگیرد. بخاطر داشته باشید که گاهی اوقات برگشتن به جایی که قبلاً بوده نیز ممکن است با شکست مواجه شود! مثالی از کارگزاری را نشان دهید که بطور موفقیت آمیز بر دوخطای حرکتی متوالی فائق آید و به هدف هم برسد.

این کارگزار مبنای خود را بر اساس واکنش فعلی که از محیط مشاهده می کند قرار می دهد و اگر بدون حسگرهایش در محیطی دچار خطا شود آنگاه حسگرهایش متوجه خطا می شوند و باعث برگشت کارگزار به محیط و از سر گرفتن مجدد عملیات می شوند.

در محیط نامعین کارگزار در نبود حسگرهایش هیچ راهی برای تعیین نتیجه واقعی ندارد بنابراین پیش آمدهای مختلف ممکن تنها چندحالت فیزیکی اضافی در حالت باور پسین هستند.

د) اکنون برای رفع خطا دو طرح مختلف را امتحان کنید: (1) به سمت نزدیکترین راس بر روی مسیر اصلی حرکت کنید؛ (2) از یک محل جدید مسیری برای رسیدن به هدف برنامه ریزی کنید. عملکرد هریک از سه طرح رفع خطا را باهم مقایسه کنید. آیا گنجاندن هزینه های جستجو در این مقایسه تاثیری دارد؟

هزینه های جستجو تعداد دفعات حرکت به حالت نهایی است که در همه یکسان نیست هرچه واحد مسافتی که طی می کنیم کمتر باشد مسلماً دارای هزینه کمتری است. و اگر این ایده که به سمت نزدیکترین راس حرکت کنیم مسافت کمتری طی می شود پس هزینه جستجو در آن تاثیر دارد.

ه) اکنون فرض کنید محلهایی وجود دارد که دارای چشم انداز یکسانی هستند. (برای مثال فرض کنید که دنیا شبکه ای است با موانع مربع شکل) کارگزار با چه نوع مسائلی مواجه است؟ راه حل ها چه شکلی هستند؟؟

چنین مسئله ای از هر مسیری که عبور کند نتایج یکسانی را خواهد داشت.

17) در متن کتاب گفتیم که مسئله هایی با هزینه ی مسیر منفی را در نظر نمی گیریم. در این تمرین آن را بیشتر شرح می دهیم:

الف) فرض کنید هزینه ی فعالیت ها مقدار منفی بزرگ است. توضیح دهید که چرا این موضوع موجب می شود هر الگوریتم بهینه ای باید کل فضای حالت را کاوش کند.

چون ممکن است در بین یک مسیر که بهینه بنظر می رسد هزینه منفی بزرگی وجود داشته باشد و روی نتیجه کل هزینه ها اثر منفی بگذارد و باعث شود آن مسیر بهینه نباشد بنابراین تمام مسیرهای ممکن موجود باید بررسی شود تا بهترین مسیر با هزینه بهینه را پیدا نمود.

ب) اگر اصرار کنیم که هزینه های مرحله ای باید بزرگتر یا مساوی یک عدد منفی ثابت مثل c باشد، کمکی به موضوع خواهد کرد؟ گراف ها و درخت هارا در نظر بگیرید.

زمانی که فضای حالت درخت است اگر هزینه ثابتی مثل عدد c را بدانیم خوب است چون باعث می شود با توجه به عمق فضای حالت، تا حدودی با توجه به عدد ثابت c هزینه ای برابر cd داشته باشیم و هزینه های بزرگتر از آن را هرس کنیم.

اما زمانی که فضای حالت گراف است خوب نیست چون در نهایت حلقه داریم و ممکن است چندین بار هزینه عدد ثابتی مثل c را دور بزنیم و تاثیری روی نتیجه نهایی ندارد.

پ) فرض کنید مجموعه ای از عملگرها وجود دارند که حلقه ای را تشکیل می دهند بطوریکه اجرای مجموعه به یک ترتیب خاص تغییری در حالت ایجاد نمی کند. اگر تمام این عملگرها هزینه ی منفی داشته باشند رفتار بهینه ی مربوط به عامل در چنین محیطی چگونه است؟

برای کارگزار موجب می شود چندین بار حلقه را دور بزند در صورتی که می توانست مسیر بهینه ای را پیدا کند.

ت) می توان حتی در حوزه هایی مانند مسیریابی، عملگرهایی را تصور کرد که دارای هزینه منفی بالایی هستند برای مثال، بعضی از پیچهای یک جاده ممکن است دارای آنچنان چشم انداز زیبایی باشند که به هزینه های معمولی زمان و سوخت بچربد. به دقت توضیح دهید که در زمینه جستجوی فضای حالت به چه علت انسانها دور یک پیچ خوش منظره به طور نامحدود (تاابد) رانندگی نمی کنند و توضیح دهید که چگونه فضای حالت و عملگرهایی برای مسیریابی تعریف کنیم که کارگزار های مصنوعی نیز از افتادن در حلقه اجتناب کنند.

یک منظره زیبا پس از 10 ها بار دیده شدن زیبایی اش را از دست می دهد در نتیجه باید برای فضای حالت یک حافظه ایجاد شود تا علاوه بر حفظ موقعیت جاری ، تعداد دفعات بازدید شده از یک مکان را هم در حافظه خود نگهداری کند تا برای میزان زیبایی یک مکان از این تابع استفاده کند.

ث) آیا می توانید یک دامنه واقعی را تصور کنید هزینه های مرحله ای منجر به حلقه شوند؟

بله مانند خوابیدن و صبح بی انگیزه بیدار شدن و رفتن به سرکار

18) دنیای جاروبرقی بدون حس گری که دارای دو محل است و تحت قانون مورفی عمل می کند را در نظر بگیرید. فضای حالت باوری که از حالت باور اولیه $\{1 و 2 و 3 و 4 و 5 و 6 و 7 و 8\}$ قابل دسترس

است را طراحی کنید و توضیح دهید که چرا این مسأله غیرقابل حل است. همچنین نشان دهید که اگر دنیا کاملاً رویت پذیر باشد آنگاه یک رشته راه حل برای هر حالت اولیه ممکن وجود دارد.

هیچ راه حلی برای این مسئله وجود ندارد زیرا مسیری وجود ندارد که به حالت باوری برسد که تمام عناصر آن شرط هدف را داشته باشد. (یک حالت باور در صورتی هدف تلقی می شود که تمام حالات فیزیکی درون آن هدف باشند) اگر مسأله کاملاً مشاهده پذیر باشد کارگزار برای رسیدن به یک حالت هدف کافی است دنباله ای اجرا کند. به عنوان مثال عملیات مکیدن آشغال فقط در یک اتاق کثیف قابل اجراست. این موضوع باعث می شود تا کارگزار رفتاری قطعی داشته باشد و به ازای هر هر حالت شروع مسأله قابل حل باشد.)

19) مسأله دنیای جاروبرقی را در نظر بگیرید

الف) کدام یک از الگوریتمهایی که در این فصل تعریف شد برای این مسأله مناسب خواهد بود؟

الگوریتمهای گراف برای این مسئله مناسبتر است چون مسیر تکراری خواهیم داشت.

آیا الگوریتم باید حالتهای تکراری را بررسی کند؟ خیر

در مقایسه با دنیای واقعی مسئله جارو برقی قابل اعتماد است و یکبار که تمیز شد دیگر هیچ وقت کثیف نمی شود یک مسأله اساسی توجه به این مطلب است که حالت هم براساس محل کارگزار و هم محلای آشغال تعیین می شود.

ب) الگوریتم منتخب خود را برای محاسبه یک رشته بهینه از اقدامات برای دنیای 3×3 که در حالت اولیه اش سه مربع بالایی کثیف است و کارگزار در مرکز قرار دارد اعمال کنید.

$24 = 3 \times 2^3$ حالت داریم که فرض شده سه مربع بالایی کثیف هستند پس کارگزار می تواند از یکی از این سه حالت شروع کند و براساس الگوریتم انتخابی کمترین هزینه مسیر را پیدا کند

یک رشته بهینه از اقدامات (عمل به چپ به راست و مکش) عبارت است از: مکش کن - به راست برو-مکش کن-به راست برو - مکش کن

ج) یک کارگزار جستجو برای دنیای جاروبرقی بسازید و کارایی آن را در مجموعه ای از دنیاهای 3×3 با احتمال 0.2 وجود آشغال در هر مربع ارزیابی کنید. با استفاده از یک نرخ تبدیل معقول، هزینه جستجو را نیز علاوه بر هزینه مسیر در معیار کارایی در نظر بگیرید.

$$0.2 = (3 \times 2^3 = 24)$$

د) بهترین کارگزار جستجوی خود را با یک کارگزار انعکاسی ساده تصادفی شده مقایسه کنید. که اگر آشغال وجود داشته باشد آن را می مکد و در غیر اینصورت به صورت تصادفی حرکت می کند.

کارگزاری که بداند کدام حالت بهینه است به آن حالت حرکت می کند ولی کارگزار تصادفی بر حسب اتفاق به یکی از حالت های موجود می رود ممکن است نسبتاً خوب عمل کند

ه) در نظر بگیرید که اگر دنیا به اندازه $n \times n$ بزرگ می شد چه اتفاقی می افتاد؟ کارایی کارگزار جستجو و کارگزار انعکاسی چگونه براساس n تغییر می کند؟

$2^n \times n$ حالت وجود دارد که بسیار زیاد است و حل این مسائل سخت است. زمان تکمیل برای یک کارگزار تصادفی رشدی کمتر از نمایی خواهد داشت.

کارایی کارگزار با توجه به n که توانی از 2 است نمایی است.