

سپاهان عادی - ۴۰۱۱۰۶۶۹۶

۱. (۲۰ نمره، درجه سختی ۸) درستی یا نادرستی عبارت‌های زیر را با ذکر دلیلی مختصر یا بیان مثال نقض مشخص کنید:

(آ) در یک فضای متناهی، درخت جستجوی می‌تواند نامتناهی باشد.

(ب) اگر آری و چهار قابل قبول باشد،  $g + \sqrt{2}g = 0.7g$  نیز حتماً قابل قبول است.

(ج) در local beam search روش انتخاب بهترین  $k$  فرزند، به انتخاب  $k$  فرزند تصادفی ارجحیت دارد و صرفاً برای انتخاب سریع تر ممکن است از روش تصادفی استفاده شود.

(د) فضای حالي وجود دارد که در آن depth-first search بدتر از روش iterative deepening search عمل کند. (مثلًا پیچیدگی زمانی  $O(n^7)$  بجزای  $O(n)$ )

(ه) برای اینکه الگوریتم BFS کامل باشد، لازم است که درجه انشتاب متناهی باشد اما برای کامل بودن الگوریتم IDS این شرط لازم نیست.

(و) جستجوی عمق اول، حالتی از Best-First می‌باشد. (یعنی تابع هزینه‌ای وجود دارد که الگوریتم Best-First را معادل کند.)

(ز) فرض کنید یک عامل  $T$  موشمند برای بازی Tetris طراحی کردیم. محیط قابل  $T$  برای این عامل، fully observable، episodic، deterministic، single agent و discrete است. (درستی یا نادرستی هر کدام از ویژگی‌های ذکر شده را بررسی کنید.)

۲) **درست** اگر از حالت‌ها<sup>۱</sup> که آنها را در جمیع اطلاعات<sup>۲</sup> نگذاریم، درخت جستجوی واند در دور (۵۰۰م)<sup>۳</sup> لی بیوصد و در تجربه درخت جستجو ناسازه خواهد بود.  
**ب) درست**

$$\left. \begin{array}{l} f \text{ is admissible} \rightarrow f \leq h^* \\ g \text{ is admissible} \rightarrow g \leq h^* \end{array} \right\} 0.7f + 0.2g \leq 0.9h^* \leq h^* \rightarrow 0.7f + 0.2g \text{ is admissible}$$

۳) **نادرست** همان‌طور که می‌دانم انجام برخی حرکات<sup>۱</sup> در حالت عادی حرکات بدی به شماری دوست (ماده انتساب تصادفی مرزیندا) می‌توانند سری برخی را به دست آورند و یا مارا از<sup>۲</sup> لیگ افتادن در local minimum<sup>۳</sup> بگیرند. **ب) درست**

۴) **درست** درختی را در تظر<sup>۱</sup> بگیرید که سری صاف است. اگر رأس هدف مادر عمق  $\ell$  باشد درخت باشد الگوریتم DFS در زمان  $O(d^\ell)$  به آن<sup>۲</sup> می‌رسد، اما الگوریتم IDS<sup>۳</sup> به صورت زیر عمل می‌کند:

$$\text{IDS time complexity: } 1 + 2 + 3 + \dots + d = \frac{d(d+1)}{2} = O(d^2)$$

عناصر  
 عن دوم  
 عن سوم  
 عن  $d^{th}$

۵) **نادرست** الگوریتم IDS همان‌طوری<sup>۱</sup> BFS تازه‌مانی<sup>۲</sup> نیک<sup>۳</sup> نیست از درخت را که این‌ها<sup>۱</sup> تکنده به این‌آزمی رسدو برای اینکه بتواند نیک<sup>۳</sup> نیست را کامل بررسی کند، با یه فریب انتساب ناسازه باشد.

۶) **درست** اگر نابع heuristic<sup>۱</sup> ای که به الگوریتم Best-First<sup>۲</sup> می‌دهیم بهمای<sup>۳</sup> گره‌های ممکن<sup>۱</sup> بابت نسبت عدد (ماته  $C = h(s_i)/h(s_j)$  که نیک نیست) دیگر<sup>۲</sup> ممکن<sup>۱</sup> که در عقیقی ستر تراز دارند عدد متری را نسبت دهد (ماته  $C = h(s_i)/h(s_j)$  که نیک نیست و دیگر<sup>۲</sup> ممکن<sup>۱</sup> که در عقیقی ستر تراز دارند عدد متری را نسبت دهد) این الگوریتم<sup>۳</sup> مانند DFS علی‌کند.  
**ز) نادرست**

-1 **fully observable** نیست: قطعات و ایزارهای بعدی بازی عالی مساعده نیستند.

-2 **single agent** نیست: باعماق دیگری در تعامل نیست.

فرموده اند

با عامل دیگری در تعامل نیست.

single agent -2

deterministic -3  
نیست: ایجاد جدید به طور تصادفی ساخته و استفاده می شوند.

episodic -4  
نیست: کسی های فعلی داری وابسته به کسی های قبلی هستند.

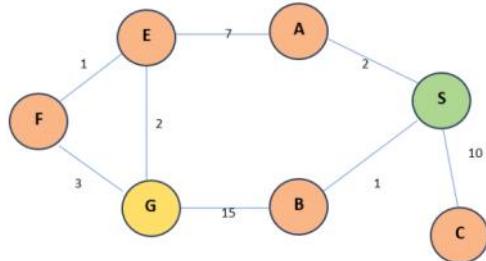
discrete -5  
نیست: محیط بازی یک محیط کسر است.

Question No.2

Saturday, March 16, 2024 11:41 PM

۱۰. نمره درجه سختی ۳) فرض کنید شکل زیر یک فضای جستجو بوده و وضعیت شروع حالت S و وضعیت پایان G باشد. اعداد نویسه شده روی پالای هزینه مسیر هستند.  
برای هر کدام از الگوریتم های زیر مشخص نماید چه مسیری طی می شود.

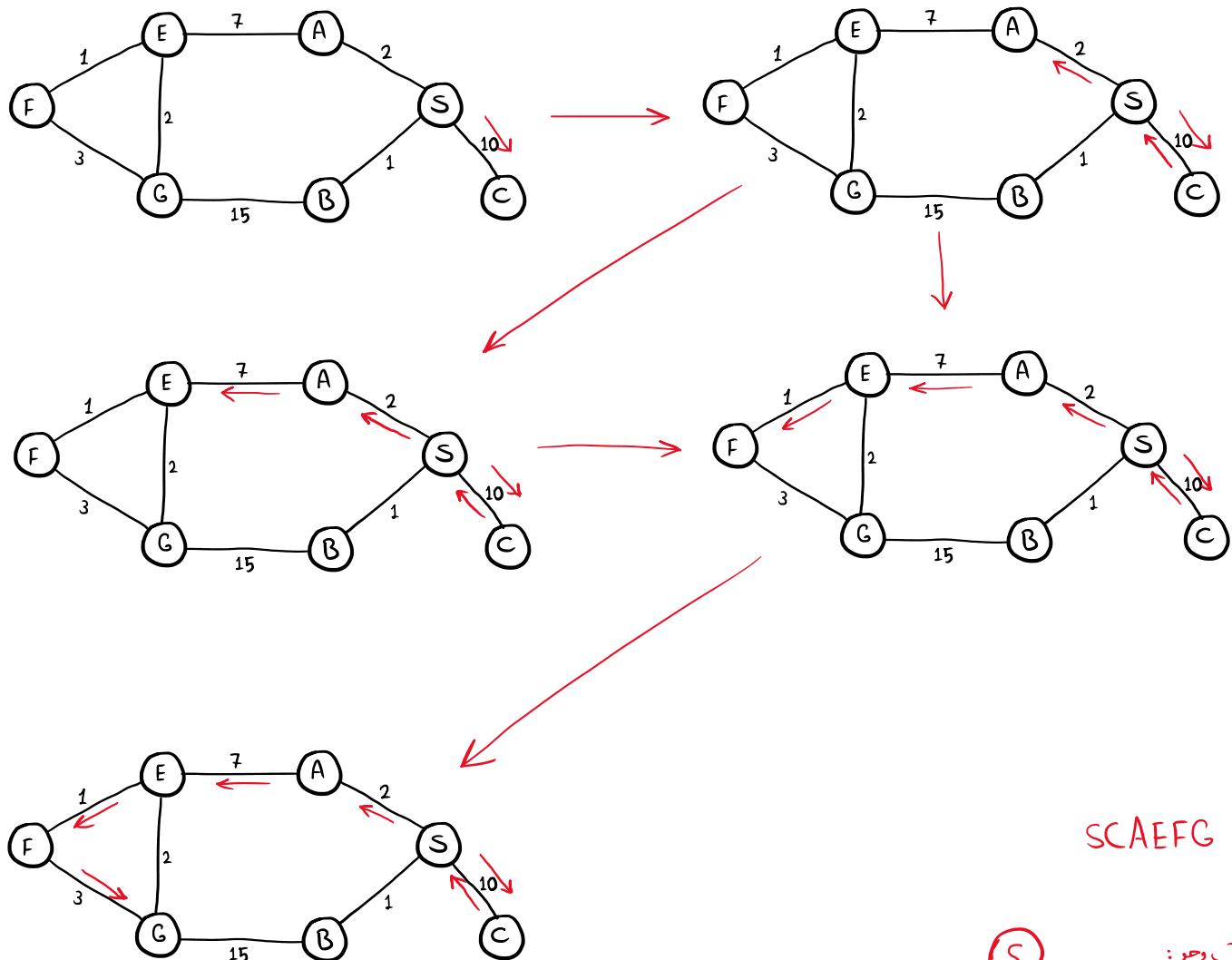
سیدیپان هدایی - 401106696

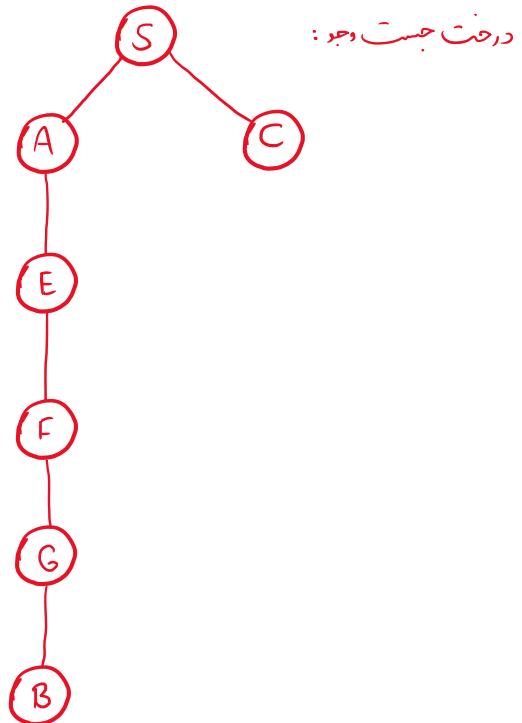
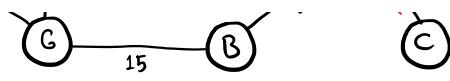


G	F	E	C	B	A	S	States
*	V	V	V	V	V	S	$h(s)$

DFS •  
greedy •  
UCS •  
 $A^*$  •  
BFS •

۱- برای الگوریتم DFS راه حلی متساوی وجود دارد که در آنها کمترین مسافت را مساعده می کنند:





- 2 - greedy: در هر مرحله node های حسابی، node ای که در آن حسینم را در fringe ذخیره کنیم، با رامارد expand می‌کنیم.

فرض: از هر مرحله، دو اس ئرایط برابر داشته باشند براساس ترتیب الفبا انتخابشان کنیم.

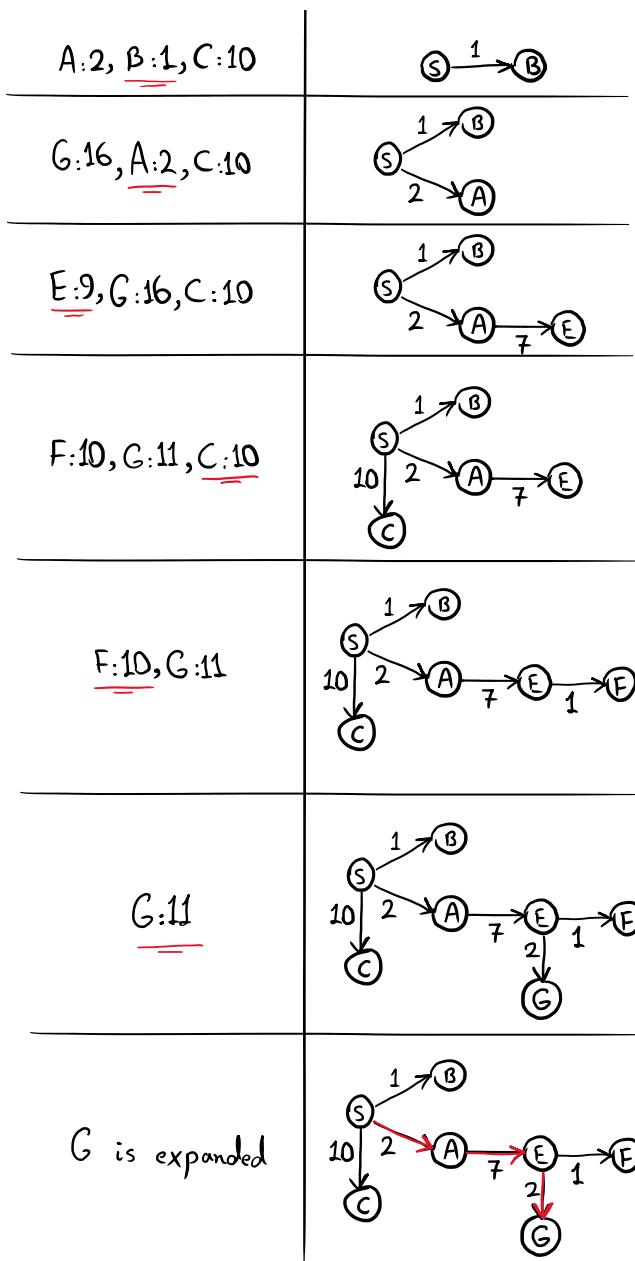
fringe	path
<u>S: 9</u>	(S)
<u>A: 7, B: 10, C: 7</u>	(S) → (A)
<u>E: 1, B: 10, C: 7</u>	(S) → (A) → (E)
<u>F: 1, G: 0, B: 10, C: 7</u>	(S) → (A) → (E) → (G)
G is expanded	(S) → (A) → (E) → (G)

زمانی دو رأس G و A در مرحله 10 از algorithm expand می‌شوند.

SAEG

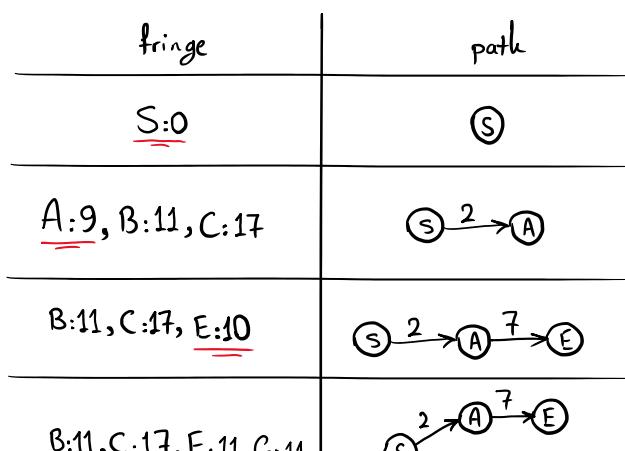
- 3 - UCS: مانند الگوریتم Dijkstra علی‌کنیم؛ در هر مرحله node های حسابی، node ای که در آن حسینم را در fringe ذخیره کنیم، با رامارد expand می‌کنیم.

fringe	path
<u>S: 0</u>	(S)
<u>A: 2, B: 1, C: 10</u>	(S) <sup>1</sup> → (B)

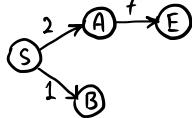


SAEG : سیر

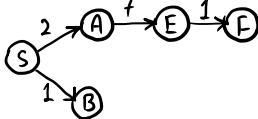
:  $(f = h + g)$  لیزد  $f$  ای دارد و  $f$  ای کمترین ذخیره ی کند و  $h$  ای  $g$  ای  $f$  ای در هر مرحله  $node$  های ممکن را در آن حسینه را در fringe ای دارد.



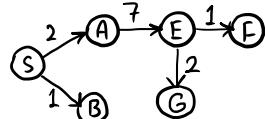
B:11, C:17, F:11, G:11



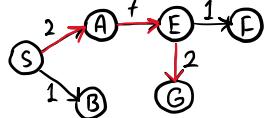
C:17, F:11, G:11



C:17, G:11

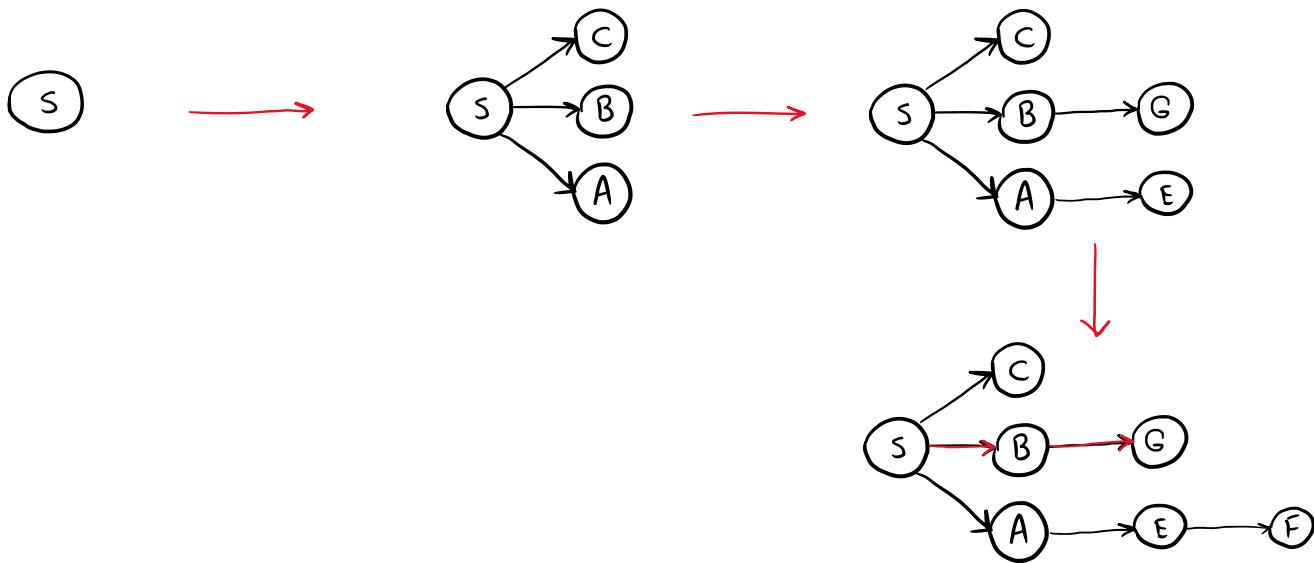


G is expanded



SAEG :

: درخت BFS به صورت زیر به دست می آید:



SBG :

Question No.3

Saturday, March 16, 2024 11:41 PM

سید کیان هدایی - 401106696

۳. ۱۵ نمره، درجه سختی ۴) الگوریتم زنگنه را در نظر بگیرید که در آن از کروموزومهای با طول ثابت ۸ و به فرم  $x = ABCDEFGH$  استفاده می‌کنند. هر زن می‌تواند عددی بین ۰ تا ۹ را بگیرد. تابع fitness را برای هر کروموزوم  $x$  به صورت زیر تعریف می‌کنند:

$$f(x) = ۳ * A + ۲ * B + C - D - E + F + ۲ * G + ۳ * H$$

جمعیت اولیه از ۴ کروموزوم زیر تشکیل شده است:

$$\begin{aligned}x_1 &= ۲۱۸۳۲۹۶۵ \\x_2 &= ۱۹۴۷۶۳۲۱ \\x_3 &= ۸۹۲۳۴۵۴۸ \\x_4 &= ۳۷۹۱۰۹۷۷\end{aligned}$$

(آ) مقدار fitness نمونه‌های داده شده را بدست آورید.

ب) حال عملیات crossover را بر روی دو نمونه با بیشترین fitness انجام دهید، به صورت تک نقطه‌ای که محل crossover نقطه میانی کروموزوم باشد. میانجین عملیات crossover را به صورت دو نقطه‌ای روی دو زمانه و سه زمانه کروموزوم از لحاظ بیشترین fitness انجام دهید. (روش این نوع crossover به این صورت است که از یک کروموزوم و بقیه زنان از یک کروموزوم دیگر بدست آید).

ج) فرض کنید دستگاهی داریم که می‌تواند یک عدد طبیعی در رفقی از ۱۰ تا ۸۹ را به صورت تصادفی خروجی بدهد. چهار بار از دستگاه استفاده کردیم و به اعداد زیر رسیدیم:

$$72, 47, 15, 75$$

حال می‌خواهیم با کمک این اعداد عملیات mutation را انجام دهیم. اگر  $\frac{۷۷}{۷۷}$  عدد دو رقمی تولید شده باشد، مقدار ۸ این حرف کروموزوم از سمت چپ را  $y$  واحد در پیمانه ۱ اضافه می‌کنیم. عملیات mutation را برای جمعیت بدست آمده در قسمت ب با توجه به اعداد دستگاه، به ترتیب از چپ به راست اعمال کنید.

د) آیا fitness جمعیت جدید نسبت به جمعیت اولیه بهتر شده است؟

ه) طبق جمعیت اولیه، آیا بدون انجام عملیات mutation امکان رسیدن به جواب بهینه (جواب با بیشترین fitness ممکن) وجود دارد؟ اگر بهله، مسیر رسیدن به آن را بگویید با تابع کمک نمکن پست.

(۱) از نسبت fitness function که در صورت سوال داده شده است اسماهی کنیم:

$$f(x) = 3A + 2B + C - D - E + F + 2G + 3H \rightarrow \begin{cases} f(x_1) = 3 \times 2 + 2 \times 1 + 8 - 3 - 2 + 9 + 2 \times 6 + 3 \times 5 \\ f(x_2) = 3 \times 1 + 2 \times 9 + 4 - 7 - 6 + 3 + 2 \times 2 + 3 \times 0 \\ f(x_3) = 3 \times 8 + 2 \times 9 + 2 - 3 - 4 + 5 + 2 \times 4 + 3 \times 8 \\ f(x_4) = 3 \times 3 + 2 \times 7 + 9 - 1 - 0 + 9 + 2 \times 7 + 3 \times 7 \end{cases}$$

$$\rightarrow \begin{cases} f(x_1) = 47 \\ f(x_2) = 19 \\ f(x_3) = 74 \\ f(x_4) = 75 \end{cases}$$

$$\left\{ \begin{array}{l} x_3 = 89234548 \\ x_4 = 37910977 \end{array} \right. \xrightarrow{\text{crossover}_1} \left\{ \begin{array}{l} y_1 = 89230977 \\ y_2 = 37914548 \end{array} \right.$$

(ب)

$$\left\{ \begin{array}{l} x_1 = 21832965 \\ x_3 = 89234548 \end{array} \right. \xrightarrow{\text{crossover}_2} \left\{ \begin{array}{l} y_3 = 21234565 \\ y_4 = 89832948 \end{array} \right.$$

(ج)

$$y_1 = 89230977 \xrightarrow{\text{mutate}(y_1, 72)} y_1 = [892309] + [(7+2) \% 10] + [7] = 89230997$$

$$y_2 = 37914548 \xrightarrow{\text{mutate}(y_2, 47)} y_2 = [379] + [(1+7) \% 10] + [4548] = 37984548$$

$$y_3 = 21234565 \xrightarrow{\text{mutate}(y_3, 15)} y_3 = [(2+5) \% 7] + [1234565] = 71234565$$

$$y_4 = 89832948 \xrightarrow{\text{mutate}(y_4, 75)} y_4 = [898329] + [(4+5) \% 10] + [8] = 89832998$$

د) برای مقایسه، fitness function را با هم مقایسه کنیم:

$$f(n) = 3A + 2B + C - D - E + F + 2G + 3H \xrightarrow{\quad} \begin{cases} f(y_1) = 3 \times 8 + 2 \times 9 + 2 - 3 - 0 + 9 + 2 \times 9 + 3 \times 7 = 89 \\ f(y_2) = 3 \times 3 + 2 \times 7 + 9 - 8 - 4 + 5 + 2 \times 4 + 3 \times 8 = 57 \\ f(y_3) = 3 \times 7 + 2 \times 1 + 2 - 3 - 4 + 5 + 2 \times 6 + 3 \times 5 = 50 \\ f(y_4) = 3 \times 8 + 2 \times 9 + 8 - 3 - 2 + 9 + 2 \times 9 + 3 \times 8 = 96 \end{cases}$$

$$\xrightarrow{\quad} \text{avg}(f(y)) = 73$$

$$, \begin{cases} f(x_1) = 47 \\ f(x_2) = 19 \\ f(x_3) = 74 \\ f(x_4) = 75 \end{cases} \xrightarrow{\quad} \text{avg}(f(x)) = 53.75$$

با توجه به اینکه میانگین fitness function نسل پنجم بزرگتر است، بدن تغییرات fitness بزرگ است.

جواب بهینه حالتی است که کروموزوم‌های متاثر با ابراز ایپ می‌باشد (H, G, F, C, B, A) که برابر با ۹ کروموزوم‌های

متاثر با ابراز ایپ می‌باشد (عنی کروموزوم‌های D و E) برابر با ۰ باشند (99900999).

با توجه به خود، crossover، متوهمی شویم که در کروموزوم جدید دم = آ، دم = ۳ (یعنی از کروموزوم‌های قبلی به است) و از به مجموعه، دم‌های اول کروموزوم‌های نسل قبلی توهجه کنیم (مجموعه {2, 1, 8, 3}). متوهمی شویم که به عقاید طبقی نباید دم اول کروموزوم‌های نسل جدید را بعد از mutation به ۶ تبدیل کرد.

سیدیکان) عدای - 401106696

۴. (۱۵ نمره، درجه سختی ۴) فرض کنید ۱۱ مهره در یک صفحه  $n \times n$  داریم. شما می توانید تمام ۱۱ مهره را هم زمان کشتل کنید. چند مهره می توانند در یک خانه قرار بگیرند و هر مهره در هر لحظه از زمان می تواند به چهت شمال، جنوب، غرب یا شرق حرکت کند و یا بنی کن. هر چیزی که مهره ای خود ثابت باقی بماند، جدول حاوی تعدادی خانه مانع است که هیچ کدام از مهره ها نمی توانند در این خانه قرار بگیرند. هدف شما این است که تمام مهره ها با کمترین مرحله ممکن در یک خانه قرار بگیرند. (در یک مرحله زمانی زمان مهره ها می توانند هم زمان با هم حرکت کنند). این مسئله را به عنوان یک مسئله مستجو در نظر بگیرید:

(الف) فرض کنید از الگوریتم  $A^*$  با کمک تابع اکتشافی برای حل این مسئله جستجو استفاده کنید. در هر مرحله زمانی  $(x_i, y_i) = p_i$  موقعیت مهره  $i$  است. برای هر کدام از توابع اکتشافی زیر، قابل قبول بودن و یکنوا<sup>\*</sup> بودن آن را مشخص کنید:

- تعداد جفت مهره هایی که در یک مکان یکسان قرار ندارند.

- اگر نزدیک ترین مهره ها فاصله حداقل ۴ داشته باشد، ۲ و در غیر اینصورت ،

$$\frac{1}{\gamma} \max_{i,j} |x_i - x_j| + \frac{1}{\gamma} \max_{i,j} |y_i - y_j|$$

$$\frac{1}{\gamma} \max(\max_{i,j} |x_i - x_j|, \max_{i,j} |y_i - y_j|)$$

(ب) فرض کنید طبق توضیحات پخش قبل، یک تابع قابل قبول و یکنوا<sup>\*</sup> برای حل مسئله داریم. حال مسئله تغییر می کند و شما  $m \leq n$  مهره دارید که از سایر مهره ها منگین شوند و بعد از هر مرحله ای که حرکت کنند، نیاز به ۱ واحد زمانی استراحت دارند (استراحت به این صورت است که هر چیزی خود می مانند). آیا می توان گفت که برای مسئله جدید همچنان  $h_0$  قابل قبول است؟ یکنوا<sup>\*</sup> جلور؟

(الف)

۱- تعداد جفت مهره های  $\geq 2$  که در یک مکان یکسان قرار ندارند:

حالی را در تظریه بینید. ۴ تا مهره داریم و این مهره ها در ۴ دو خانه  $\geq 2$  دو تایی در دو خانه  $\geq 1$  دو تایی دارند. در این حالت با یک حرکت یک توان به حالت زیر رسید  $(h(state) = 1)$ . اما حاصل تابع اکتشافی برابر است با:

$$\rightarrow h(state) > h^*(state) \leftarrow$$

← این تابع اکتشافی نه قابل قبول است و نه یکنوا<sup>\*</sup>.

۲- اگر نزدیک ترین مهره صاحفه اصله حداقل  $\geq 4$  داشته باشد،  $\frac{1}{2}$  و در غیر اینصورت  $\geq 0$ :

اگر حداقل مفاصله هر دو صاحفه ای  $\geq 4$  باشد، حداقل به  $\lceil \frac{l}{2} \rceil$  حرکت نیاز داریم (هر دو مهره به ممتجم حرکت می کنند). با توجه به تعریف تابع دوی  $\lceil \frac{l}{2} \rceil$  حالت بنی می کنیم:

$$\left. \begin{array}{l} l \geq 4 \rightarrow h^*(state) \geq \lceil \frac{l}{2} \rceil = 2, h(state) = 2 \\ l < 4 \rightarrow h^*(state) \geq \lceil \frac{l}{2} \rceil \geq 0, h(state) = 0 \end{array} \right\} h^*(state) \geq h(state) \rightarrow h \text{ is admissible}$$

همانطور که دیدیم تابع قابل قبول است. حال یک ذایی آن را بررسی می کنیم:

دو حالت محاسبه را در تظریه بینید که در یکی  $\frac{l}{2} = 3$  و در دیگری  $\frac{l}{2} = 0$  است. بنابراین داریم:

$$\begin{cases} l_1 = 3 \rightarrow h_1 = 0 \\ l_2 = 5 \rightarrow h_2 = 2 \end{cases} \quad h_2 - h_1 = 2$$

می‌دانیم که هر یکی از این احتمالات بحالات دیگر نیست. بنابراین:

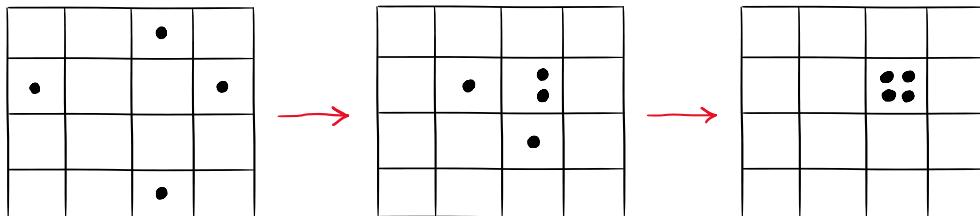
$$w_{s_1 \rightarrow s_2} = 1 \rightarrow h_2 - h_1 > w_{s_1 \rightarrow s_2} \rightarrow h \text{ is not monotonic}$$

این تابع قابل قبول است و نیک نواست.

$$:= \frac{1}{2} \max_{i,j} |x_i - x_j| + \frac{1}{2} \max_{i,j} |y_i - y_j| - 3$$

جدول زیر را در ذطر بخیرید:

$$h^*(\text{state}) = 2$$



$$h(\text{state}) = \frac{1}{2} \times 3 + \frac{1}{2} \times 3 = 3 \rightarrow h(\text{state}) > h^*(\text{state})$$

این تابع دسته دوی قابل قبول است و نیک نوا.

$$:= \frac{1}{2} \max \left( \max_{i,j} |x_i - x_j|, \max_{i,j} |y_i - y_j| \right) - 4$$

مسنون است و دووجهی که فاصله Manhattan، یعنی  $\sqrt{2}$  است، حداقل باید  $\lceil \frac{d}{\sqrt{2}} \rceil$  حرکت انجام بدهند تا به هم برسند. از سینه فاصله Manhattan بروهارا در تقریب بزرگ، خواهیم داشت:

$$l = \max_{i,j} (|x_i - x_j| + |y_i - y_j|) \rightarrow \max_{i,j} |x_i - x_j| \leq l, \max_{i,j} |y_i - y_j| \leq l$$

$$\rightarrow \max \left( \max_{i,j} |x_i - x_j|, \max_{i,j} |y_i - y_j| \right) \leq l \rightarrow \frac{1}{2} \max \left( \max_{i,j} |x_i - x_j|, \max_{i,j} |y_i - y_j| \right) \leq \frac{l}{2}$$

$$, \frac{l}{2} \leq \lceil \frac{l}{2} \rceil, h^*(\text{state}) \geq \lceil \frac{l}{2} \rceil$$

$$\rightarrow h(\text{state}) \leq h^*(\text{state}) \rightarrow h \text{ is admissible}$$

$$h(state) \leq h^*(state) \rightarrow h \text{ is admissible}$$

حال که دیگر این تابع قابل قبول است، یک توی آن را بررسی کنیم:

اگر دو حالت ممکن برای در تظری هستیم که سمتین فاصله' Manhattan درین لئے  $\frac{l}{2}$  و در دیگری  $\frac{l-2}{2}$  باشد (دفرانسیل هزینه های  $\frac{1}{2}$  و  $\frac{1}{2}$  می توانند به اندازه'  $\frac{1}{2}$  تغییر کند که این به معنای این است که فاصله' Manhattan حداقل به اندازه'  $\frac{1}{2}$  تغییر کند) حواهیم داشت:

$$h(state_1) = \frac{l}{2}, h(state_2) > \frac{l-2}{2} \rightarrow |h(state_2) - h(state_1)| \leq \frac{l}{2} - \frac{l-2}{2} = 1 \rightarrow |h(state_2) - h(state_1)| \leq 1$$

و همانطور که می دانیم هزینه' متن از یک حالت به حالت دیگر برابر با ۱ است:

$$w_{s_1 \rightarrow s_2} = 1 \rightarrow |h(state_2) - h(state_1)| \leq w_{s_1 \rightarrow s_2} \rightarrow h \text{ is monotonic}$$

با این تابع  $h$  ممکن قابل قبول دیگر نیست.

ب)  $h$  admissibility تابع همچنان باقی می باشد، جزو که مسئله صرفه کی تغییرات را شده و این شرط که تعدادی از همه حالت های از مرحله اول که مسئله اضافه شده است. می باند و در حالتی که  $m=0$  باشد، مسئله همچنان تغییرات ندارد است. در حالت های دیگر عملکرد این هزینه افزایش پیدا نماید. حالات راهی (هدف) برسیم.

تابع  $h$  تغییراتی نمی کند. با اینجا این تغییرات در مسئله، در پیشین حالت همچنان و مسئله همانطور که بود باقی می باشد و در حالت های دیگر، اعمال این دهزینه' اضافی (نسبت به حالت قبلی مسئله) می شود که در این صورت باید با موج هزینه' پیشتر از حالتی به حالت دیگر برویم که این قضیه به monotonicity مسئله مملکتی کند.

سیدیان هدایی - 401106696

۵. ۱۵ نمره، درجه سختی ۵) یک جدول  $n \times n$  داریم که هر خانه‌ی آن سیاه یا سفید است. در مرحله می‌توانیم یک زیرجدول از آن را انتخاب کنیم و رنگ خانه‌های درون آن را معکوس کنیم (از سیاه به سفید و بالعکس).

می‌خواهیم با کمترین تعداد مرحله کاری کنیم که جدول کاملاً سفید شود.

(الف) فضای مساله را گوئی‌ای پیکربندی کنید به نحوی که حالت‌ها، کشش‌ها، ضریب انتساب و حالت اولیه و

نهایی مساله واضح باشد و آنها را مشخص کنید.

(ب) اندازه فضای مساله را به فرمت O بزرگ و بر حسب  $n$  بدست آورید.

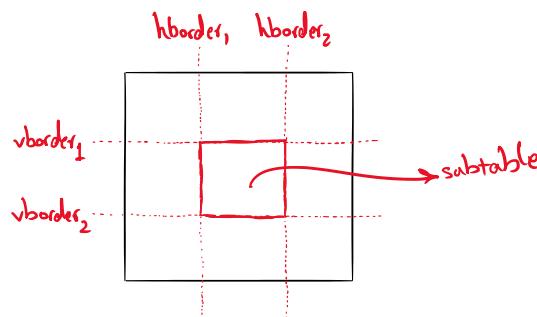
(ج) یک تابع اکتشافی تابیه‌ی برای حل این مساله ارائه دهید و این تابع را از لحاظ یکنوانی و قابل قبول بودن بررسی کنید.

**حالات**: هر حالت  $S^n$  دهنده یک جدول  $n \times n$  است که هر خانه‌ی آن یاری سفید است و یا سیاه. ماهر حالت را با یک آرایه دو بعدی  $S^n$  نشان می‌دهیم که معکوس آن یا شامل عرف  $W^n$  است (یعنی یک سفید) و یا حرف  $B^n$  (یعنی یک سیاه).

**لکس**: انتساب یک زیرجدول و تغییر رنگ خانه‌های آن زیرجدول (تبیل  $B^n$  به  $W^n$  و تبیل  $W^n$  به  $B^n$ ).

**ضریب انتساب**: همانطور که می‌دانیم ضریب انتساب برابر با عدد ادکنش‌های ممکن (تعداد حالت‌های انتساب زیرجدول) در هر حالت است.

با توجه به اینه در راستای عمودی و افقی جدول  $n \times n$  خانه داریم،  $n+1$  حالت برای انتساب مرزهای عمودی یا افقی داریم:



$$\text{branching factor} = \binom{n+1}{2} \times \binom{n+1}{2} = \binom{n+1}{2}^2$$

کل تعداد حالت های ممکن ب مرزهای افقی  
کل تعداد حالت های ممکن ب مرزهای عمودی

**حالات اولیه**: آرایه دو بعدی  $S^n$  دهنده جدول و دو دیگر.

**حالات نهایی**: آرایه دو بعدی  $S^n$  با خانه‌های  $W^n$  (تمام سفید).

ب) هر جدول دارای  $\frac{n^2}{2}$  خانه است و هر خانه هم می‌تواند سیاه یا سفید باشد ( $\frac{n^2}{2}$  حالت دارد).

بنابراین دارای  $\frac{n^2}{2}$  حالت مختلف هستیم:

$$O(2^{n^2}) = \text{اندازه فضای مساله}$$

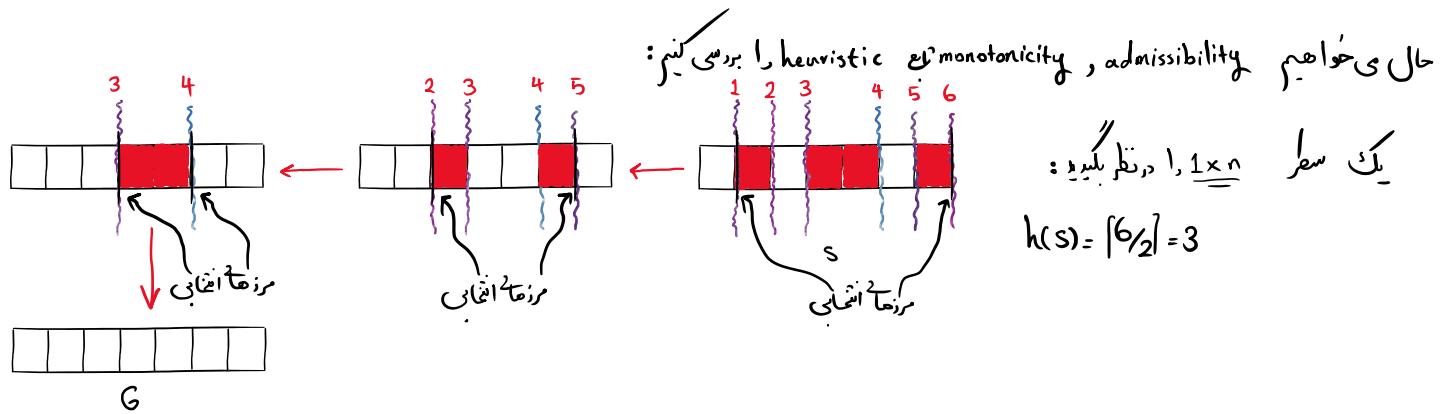
ج) تابع **heuristic** را به صورت زیر تعریف کنیم:

به ازای هر سطح و به ازای هر مطابق اد مرزهای که دو طرفی خانه‌ای باشند، مقدار ممکن و مقدار ممکن آنها را با  $\frac{N}{2}$  مساله.

تابع **heuristic** را برابر با  $\frac{N}{2}$  مساله کناریم:

$$h(\text{state}) = \left\lceil \frac{N}{2} \right\rceil$$

حالی خواهیم **admissibility** و **monotonicity** **heuristic** را بررسی کنیم:



همانطور که بیشتر، حداقل دو تا از مرزها در هر سفر یا شون (عملن اینست مطابق با همین دلای از مرزها تغییر نمی‌کند) در هر لشکر تغییری کنند:

$$|h(state_1) - h(state_2)| = \left\lceil \frac{N}{2} \right\rceil - \left\lceil \frac{N-2}{2} \right\rceil = \left\lceil \frac{N}{2} \right\rceil - \left\lceil \frac{N}{2} - 1 \right\rceil \leq 1$$

با توجه به اینه هزینه، رضن از یک حالت به حالت دیگر (وزن یا  $s_1 \rightarrow s_2$ ) برابر با ۱ است داریم:

$$w_{s_1 \rightarrow s_2} = 1 \rightarrow |h(state_1) - h(state_2)| \leq w_{s_1 \rightarrow s_2} \rightarrow \text{heuristic function is } \underline{\text{monotonic}}$$

حال با توجه به اینکه تابع monotonic، heuristic شیوه‌ی کمترین admissible هست.

Question No.6

Saturday, March 16, 2024 11:41 PM

سید کیان عدایی - ۴۰۱۱۰۶۶۹۶

۶. نمره، درجه سختی ۶) به سوالات زیر درباره الگوریتم A\* پاسخ دهید:

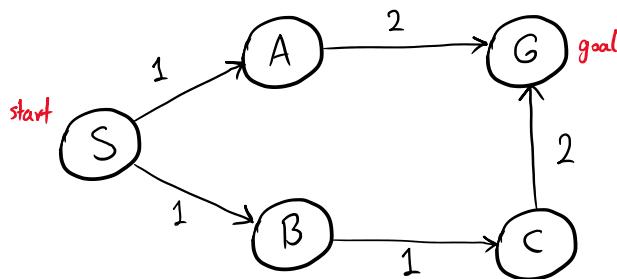
- ۱. یک مثال از یک گراف جهت دار و یک تابع heuristic نه لزوما قابل قبول بینید که در آن الگوریتم A\* مسیر بهینه را پیدا نکند. هزینه تمامی بالها باید مثبت باشد و گراف حداکثر ۶ گره داشته باشد.

گره های شروع و پایان را مشخص کنید و برای گره های باقی مقدار تابع heuristic را بنویسید.

- آ) مسیر بهینه را مشخص نمایید.

- ب) مسیری که A\* بیندازد را مشخص نمایید.

- ۲. سوال بالا را برای گراف جهت دار بدون دور و یک تابع heuristic قابل قبول و نه لزوما بیندازد حداکثر ۸ گره حل نمایید.

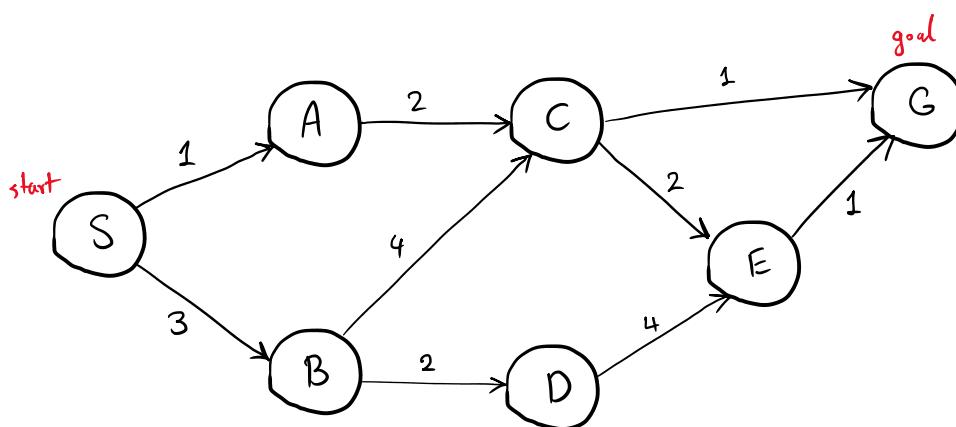


$$h(S) = 3, h(A) = 13, h(B) = 3, h(C) = 8, h(G) = 0$$

مسیر بهینه: SAG با طول ۳

fringe	path
<u>S: 3</u>	(S)
A: 14, <u>B: 4</u>	(S) → (B)
A: 14, <u>C: 10</u>	(S) → (B) → (C)
A: 14, <u>G: 4</u>	(S) → (B) → (C) → (G)
G is expanded	(S) → (B) → (C) → (G)

مسیر: SBCG با طول ۴



$$h(S) = 8, h(A) = 17, h(B) = 10, h(C) = 13, h(D) = 7, h(E) = 1, h(G) = 0$$

مسیر: SACG با طول ۴

لطفاً monotonic smooth heuristic است:

$$\left. \begin{array}{l} f(A) = h(A) + g(A) = 17 + 1 = 18 \\ f(C) = h(C) + g(C) = 13 + 3 = 16 \end{array} \right\} f(A) > f(C)$$

$h$  is not monotonic  $\leftarrow$

حالی خواهیم سیری به این شده توسط الگوریتم  $A^*$  را بینم:

مسیری اشده توسعه 10 باطل  $A^* SBDEC : A^*$

fringe	path
<u><math>S: 8</math></u>	(S)
$A: 18, B: 13$	(S) $\rightarrow$ (B)
$A: 18, C: 20, D: 12$	(S) $\rightarrow$ (B) $\rightarrow$ (D)
$A: 18, C: 20, E: 10$	(S) $\rightarrow$ (B) $\rightarrow$ (D) $\rightarrow$ (E)
$A: 18, C: 20, G: 10$	(S) $\rightarrow$ (B) $\rightarrow$ (D) $\rightarrow$ (E) $\rightarrow$ (G)
$G$ is expanded	(S) $\rightarrow$ (B) $\rightarrow$ (D) $\rightarrow$ (E) $\rightarrow$ (G)