بسمه تعالی

دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پاسخ تمرین سری دوم مبانی و کاربردهای هوش مصنوعی

«فصل سوم»

نیم‌سال دوم ۱۳۹۹-۱۳۹۸

۱- بازی word ladder به این شکل است که در ابتدا دو کلمه n حرفی انتخاب می‌نماییم و می‌خواهیم از کلمه اول به کلمه دوم برسیم به این صورت که در هر مرحله یک حرف از کلمه مرحله قبل را تغییر می‌دهیم تا در نهایت به کلمه نهایی برسیم:

الف) این مسئله را به صورت یک مسئله جستجو فرموله‌بندی نمایید و فاکتور انشعاب را بیابید.

پاسخ)

حالات: تمام کلمات ساخته‌شده با جابه‌جایی حروف کلمه اول

حالت اولیه: کلمه اول

اعمال: در هر مرحله، تغییر یکی از حروف کلمه

آزمون هدف: آیا کلمه n حرفی همان کلمه دوم است؟

هزینه مسیر: تعداد تغییر حروف انجام شده برای تغییر کلمه اول به کلمه دوم

حداکثر ضریب انشعاب برابر است با:

$$\text{تعداد حروف الفبای انگلیسی} \times \text{تعداد حروف کلمه} = 26 \times n$$

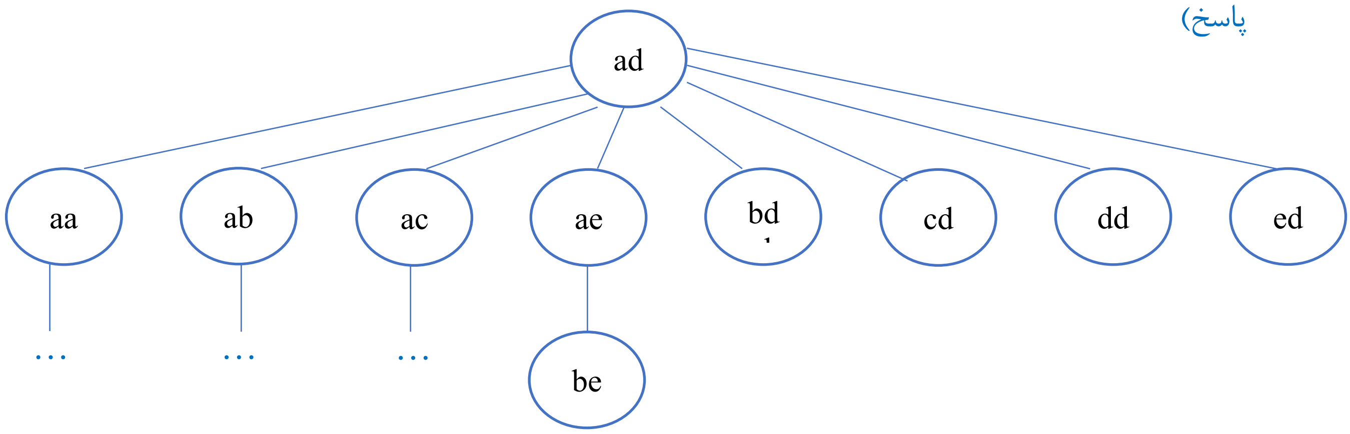
ب) برای حل این مسئله چه روشی پیشنهاد می‌دهید؟

پاسخ)

تغییر حروف در کلمات را می‌توان با یک گراف نشان داد. به این صورت که هر گره‌ی گراف یک کلمه را نشان دهد و هر یال گراف، یک انتقال از یک کلمه به کلمه دیگر را با تغییر حروف آن کلمه نشان دهد. بنابراین هدف پیدا کردن کوتاه‌ترین مسیر گراف، بین کلمه اول و کلمه دوم است. برای حل مسئله می‌توان از BFS استفاده کرد و چون هزینه تمامی اعمال یکسان است، جواب بهینه را می‌دهد.

ج) در صورتی که n=2 و کلمه ابتدایی ad و انتهایی be باشد گراف جستجو را رسم کنید. (فرض کنید فقط از حروف a تا e استفاده می‌شود.)

پاسخ)



۲- مسئله‌ی حرکت k اسب از k مربع ابتدایی $s_1, \dots, s_k$ به k مربع هدف $g_1, \dots, g_k$ در یک فضای شطرنج نامحدود را درنظر بگیرید. این حرکت شامل این قانون می‌شود که هیچ دو اسبی نباید در یک زمان در یک مربع حضور داشته باشند. هر عمل شامل حرکت دادن حداکثر k اسب به‌صورت همزمان میباشد. هدف ما حل این مسئله با کمترین تعداد اعمال می‌باشد.

الف) ماکزیمم ضریب انشعاب b در این فضای جستجو چیست؟

پاسخ)

برای هر کدام از k اسب ۸ حرکت به علاوه‌ی یک حالت که آن اسب حرکت نکند، وجود دارد. پس در مجموع ۹ حرکت برای هر اسب وجود دارد. پس برای حرکت k اسب، ماکزیمم ضریب انشعاب برابر با 9^k است.

ب) فرض کنید h_i یک هیوریستیک قابل قبول برای مسئله‌ی حرکت اسب iام به هدف g_i به‌تنهایی باشد. کدام یک از هیوریستیک‌های زیر برای مسئله‌ی k اسب قابل قبول است؟

$$\sum_{i=1}^{k} h_i \qquad \max\{h_1, \dots, h_k\} \qquad \min\{h_1, \dots, h_k\}$$

پاسخ)

اگر مسئله را به مسئله‌ی تعدیل‌شده تبدیل کنیم به این صورت که اسب‌ها بتوانند در یک زمان در یک مربع قرار بگیرند، آنگاه $\max\{h_1, \dots, h_k\}$ یک تابع هیوریستیک برای این مسئله خواهد بود و چون h_iها قابل‌قبول هستند، یعنی کمتر از هزینه‌ی واقعی هستند پس $\max\{h_1, \dots, h_k\}$ نیز قابل قبول است.

تابع $\min\{h_1, \dots, h_k\}$ مقداری کمتر یا مساوی با $\max\{h_1, \dots, h_k\}$ دارد، پس این تابع نیز قابل‌قبول است.

تابع $\sum_{i=1}^{k} h_i$ قابل قبول نیست. حالتی را در نظر بگیرید که هر g_i تنها یک خانه با s_i متناظر آن فاصله دارد. در این حالت هزینه بهینه ۱ است در حالیکه تابع $\sum_{i=1}^{k} h_i$ مقدار k را بر می‌گرداند.

ج) کدام یک از هیوریستیک‌ها بهترین است؟

$$\sum_{i=1}^{k} h_i \qquad \max\{h_1, \dots, h_k\} \qquad \min\{h_1, \dots, h_k\}$$

پاسخ)

تابع $\max\{h_1, \dots, h_k\}$ بر تابع $\min\{h_1, \dots, h_k\}$ غلبه می‌کند، پس تابع بهتری است. تابع $\sum_{i=1}^{k} h_i$ به دلیل اینکه سازگار نیست، تابع خوبی نیست.

۳- اتاقی را در نظر بگیرید که ابعاد آن،۲ در ۲ در ۳ متر است. از ۴ گوشه‌ی سقفِ ۳ متری اتاق، ۴ موز آویزان است و میمونی که قد یک متر دارد می‌خواهد با دو صندوق یکسان مکعبی شکلِ ۱ در ۱ در ۱ متری که در اختیار دارد، موزها را بردارد. این صندوق‌ها متحرک، قابل بالا رفتن و قابل روی هم چیدن هستند. تعداد کل حالات در فضای حالت چند است؟ توضیح دهید. حالت اولیه، آزمون هدف و تابع هزینه مسیر را نیز مشخص نمایید

پاسخ)

حالات چیدمان این دو صندوق ۱۰ است، به ازای هر چیدمان، میمون می‌تواند در هر یک از ۴ خانه باشد و نیز به ازای هر چیدمان هر یک از موزها می‌تواند چیده یا همچنان آویزان باشد. بنابراین تعداد کل حالات برابر خواهد بود با $۱۰ \times ۴ \times ۲^{۴}$

حالت ابتدایی برابر است با هر کدام از حالاتی که هر ۴ موز آویزان است (۴۰ حالت).

آزمون هدف برابر است با حالاتی که تمامی چهار موز چیده شده باشد (۴۰ حالت) - از این ۴۰ حالت بعضی حالات غیرقابل دسترس هستند (حالت قبل این حالات، حالت هدف بوده و هیچ گاه این حالات در فضا بوجود نخواهند آمد) برای مثال حالتی که موزها همگی چیده شده و دو صندوق روی هم قرار ندارند (این حالات را نیز می توان از تعداد کل حالات کم کرد و تعداد دقیق حالات را بدست آورد).

تابع هزینه مسیر: تعداد حرکات انجام شده

۴- فرض کنید دو دوست در دو شهر متفاوت زندگی می‌کنند هدف نهایی آن است که در کوتاه‌ترین زمان ممکن این دو فرد یکدیگر را ملاقات کنند. برای این کار در هر مرحله این دو نفر به صورت هم‌زمان می‌توانند از شهری که در آن قرار دارند به شهر همسایه آن شهر عزیمت کنند، با این شرط که اگر یکی از دو دوست زودتر به شهر همسایه مورد نظرش رسید، بایستی تا رسیدن دوست خود به شهر همسایه‌ای که قصد عزیمتش را داشت صبر کند تا مرحله بعدی مسافرت آغاز شود. مقدار زمان مورد نیاز برای رفتن از شهر *i* به شهر همسایه *j* برابر با مسافت *d(i,j)* بین دو شهر است.

الف) این مسأله را به صورت یک مساله جستجو فرموله کنید.

پاسخ)

حالات: تمام جفت شهرهای ممکن بصورت (a , b)

حالت اولیه: دو شهری که این دو نفر در آن قرار دارند.

اعمال: اگر این دو نفر در شهرهای a و b باشند (a , b)، اعمال ممکن عبارتند از تمام جفت شهرهایی که به شکل (x, y) هستند بطوریکه x شهر مجاور a و y شهر مجاور b است.

آزمون هدف: بودن در مکان (i, i) برای یک i مشخص

هزینه مسیر: هزینه رفتن از وضعیت (i, j) به وضعیت (n, m) برابر است با: max(d(i, n) , d(j, m))

ب) آیا امکان دارد کوتاه‌ترین راه این باشد که ملاقات دو دوست در محل اقامت یکی از آنان باشد؟

پاسخ)

اگر فرض کنیم یکی از nodeهایی که این دو نفر در آن اقامت دارند دارای loop باشد، در اینصورت اگر تنها ۲ شهر وجود داشته باشد یا تعداد شهرهای بین این ۲ نفر، عدد فردی باشد، کوتاه‌ترین راه، محل اقامت یکی از آنان است.

ج) یک تابع هیوریستیک سازگار برای این مسأله پیشنهاد دهید.

پاسخ)

برای سازگار بودن یک هیوریستیک، باید رابطه زیر برای هر گره n و پسین آن مانند n’ برقرار باشد:

$h(n) \le c(n,a,n') + h(n')$

فرض کنید D(n, g) فاصله اقلیدسی بین شهر n و شهر هدف باشد. طبق نامساوی مثلثی، هزینه رفتن بصورت مستقیم از شهر n به شهر هدف کمتر از هزینه‌رفتن به شهر میانی n' و سپس از شهر n' به شهر هدف است:

$D(n, g) \le D(n, n') + D(n', g)$

در بهترین حالت، هر دو نفر بصورت مستقیم به سمت یکدیگر حرکت می‌کنند(یعنی حرکات اضافه‌ای ندارند) که در نتیجه فاصله بین آن‌ها در هر مرحله نصف هزینه مسیر می‌شود. پس می‌توان گفت که D(a, b)/2 یک هیوریستیک سازگار است زیرا:

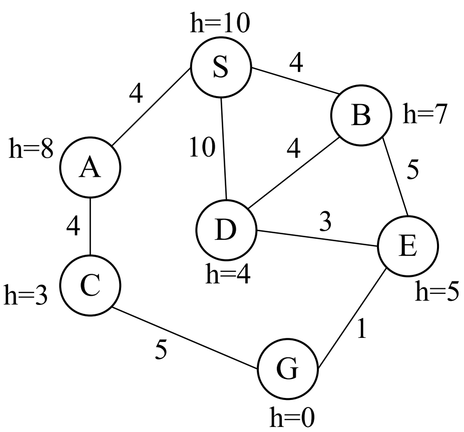
$D(n, g) \le D(n, n') + D(n', g) \longrightarrow D(n, g)/2 \le D(n, n')/2 + D(n', g)/2 \longrightarrow$

$h(n) \le D(n, n')/2 + h(n') \longrightarrow h(n) \le D(n, n') + h(n') \longrightarrow h(n) \le c(n, a, n') + h(n')$

د) آیا حالتی از چیدمان شهرها روی یک نقشه وجود دارد که این دو نفر هیچگاه به هم نرسند و مسئله راه‌حلی نداشته باشد؟

پاسخ)

نقشه‌ای را در نظر بگیرید که تنها دو شهر روی آن وجود دارد. در این حالت این دو نفر دائماً در حال جابه‌جا کردن مکان‌هایشان با یکدیگر هستند و هیچگاه در یک شهر قرار نمی‌گیرند.



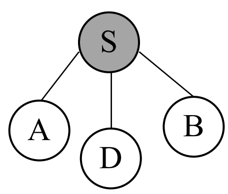
۵- شکل روبه‌رو، مسئله‌ی جستجویی را نشان می‌دهد که به‌صورت گراف مدل شده است. وضعیت شروع S بوده و تنها وضعیت هدف G است. اعداد نشان داده‌شده بر روی یال‌ها هزینه‌ی هر عمل را نشان می‌دهند. (توجه: در اجرای هر یک از الگوریتم‌های گفته‌شده در ادامه، در صورتِ وجودِ شرایط یکسان براساس حروف الفبا عمل کنید.)

الف) مسیر برگردانده‌شده توسط الگوریتم جستجوی گرافی UCS چیست؟ تغییرات مجموعه‌های مرزی و کاوش‌شده را در هر مرحله نمایش دهید.

پاسخ) مسیر S-B-E-G

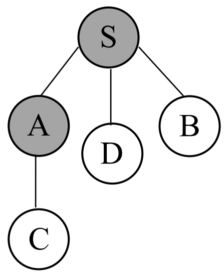
Frontier = {S}

Explored = {}



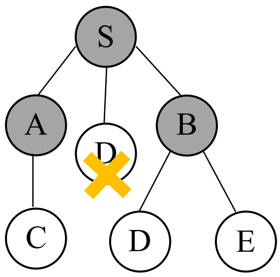
Frontier = {A, B, D}

Explored = {S}



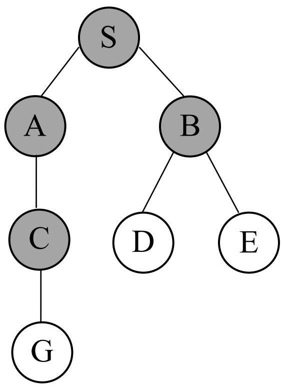
Frontier = {B, D, C}

Explored = {S, A}



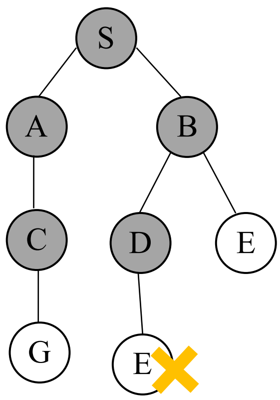
Frontier = {C, D, E}

Explored = {S, A, B}



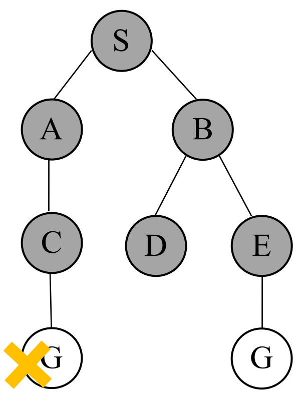
Frontier = {D, E, G}

Explored = {S, A, B, C}



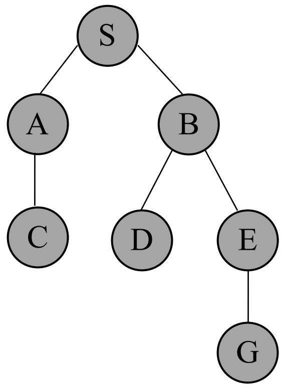
Frontier = {E, G}

Explored = {S, A, B, C, D}



Frontier = {G}

Explored = {S, A, B, C, D, E}

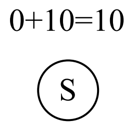


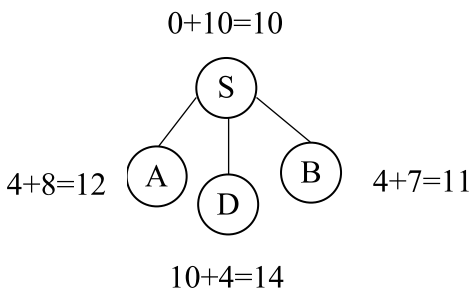
Frontier = { }
Explored = {S, A, B, C, D, E, G}

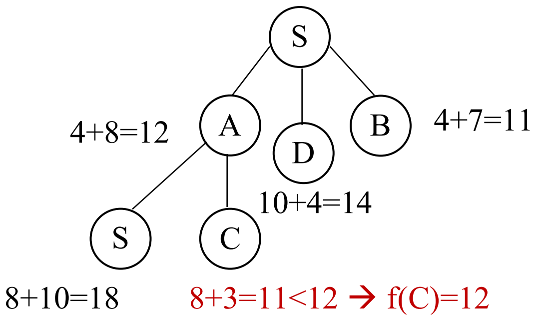
Frontier	Explored
S	
A(4), B(4), D(10)	S
B(4), D(10), C(8)	S, A
D(8), C(8), E(9)	S, A, B
D(8), E(9), G(13)	S, A, B, C
E(9), G(13)	S, A, B, C, D
G(10)	S, A, B, C, D, E
	S, A, B, C, D, E, G

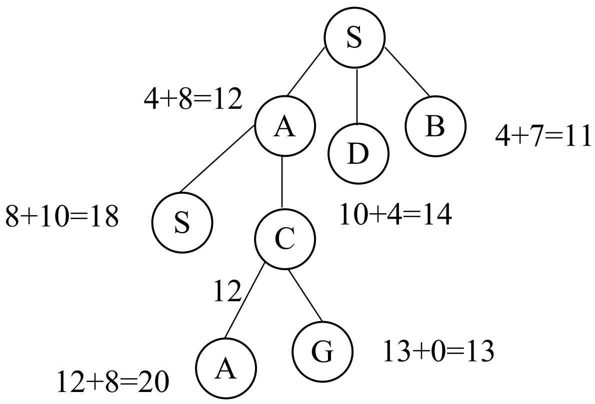
ب) الگوریتم *IDA را برای آستانه ۱۲ اجرا کنید.

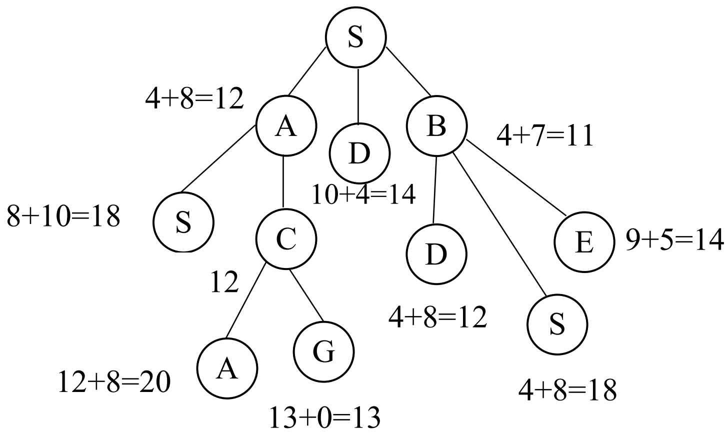
پاسخ)

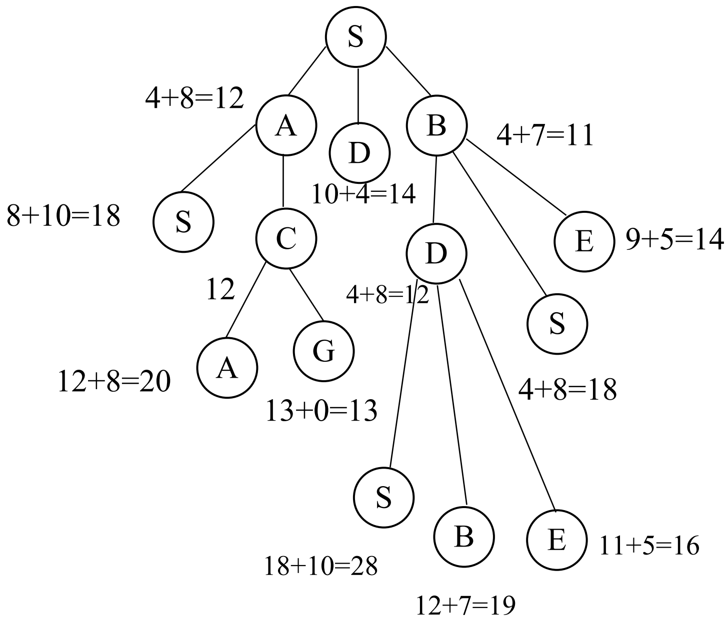





S
A
D
B
S
C
4+8=12
4+7=11
10+4=14
8+10=18
8+3=11<12 → f(C)=12


S
A
D
B
S
C
A
G
4+8=12
4+7=11
8+10=18
10+4=14
12
12+8=20
13+0=13


S
A
D
B
S
C
D
E
S
A
G
4+8=12
4+7=11
10+4=14
8+10=18
9+5=14
12
4+8=12
12+8=20
4+8=18
13+0=13

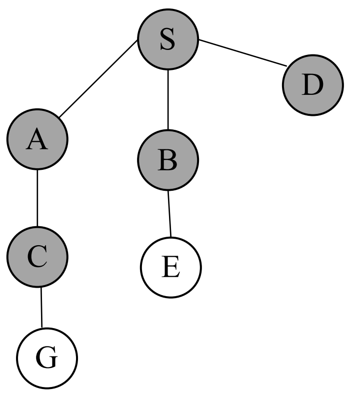


ج) کارایی دو الگوریتم جستجوی <u>گرافی</u> BFS و DFS را از نظر <u>تعداد گره تولید شده</u> تا رسیدن به هدف برای این مسئله مشخص کنید.

پاسخ)

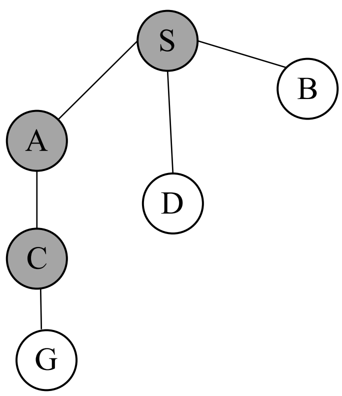
BFS ۷ گره تولید می‌کند.

آزمون هدف: هنگام تولید نود

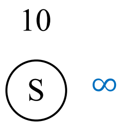


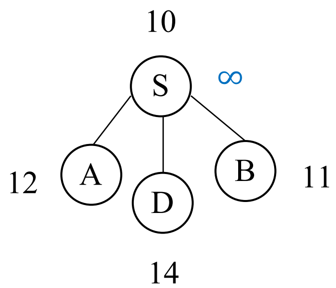
DFS ۶ گره تولید می‌کند.

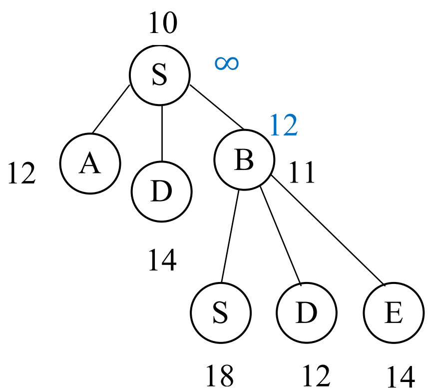
آزمون هدف: هنگام بسط نود

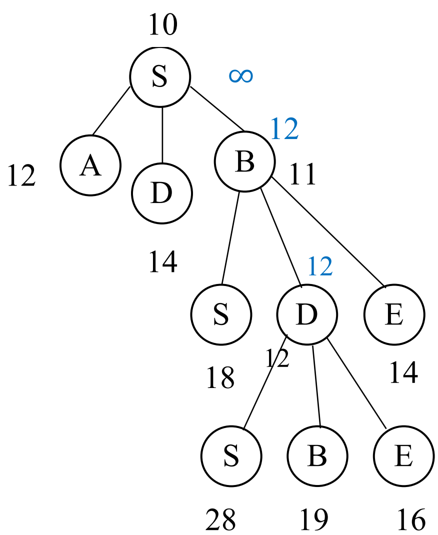


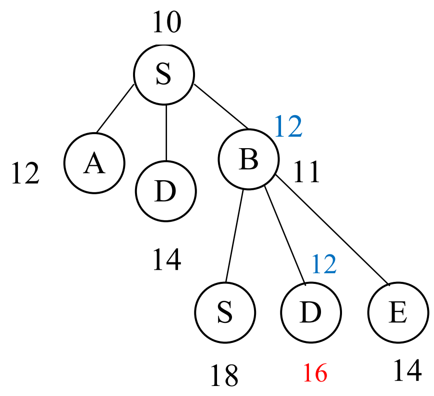
د) ترتیب تولید و گسترش گره‌ها را با استفاده از روش RBFS مشخص کنید.
پاسخ)



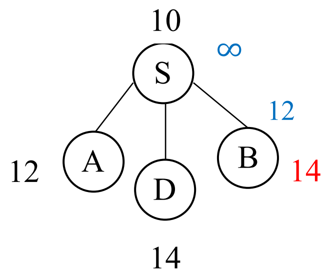




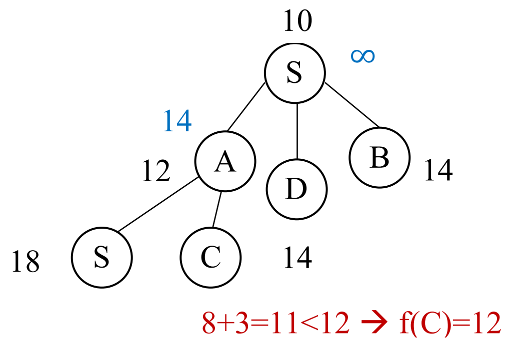




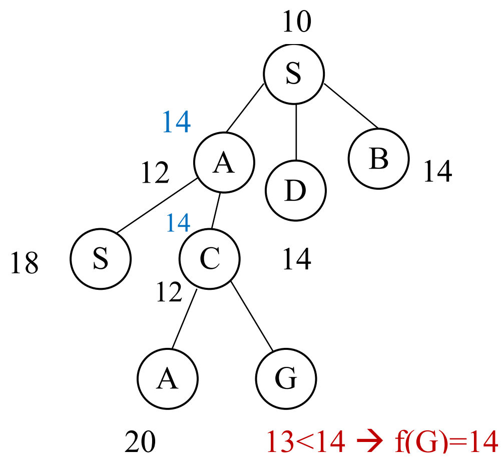
10
S
12
A
D
B
11
12
14
12
S
D
E
18
16
14



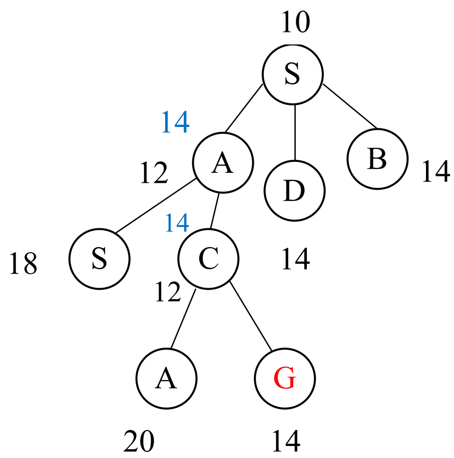
10
S
∞
12
12
A
D
B
14
14



10
S
∞
14
12
A
D
B
14
18
S
C
14
8+3=11<12 → f(C)=12

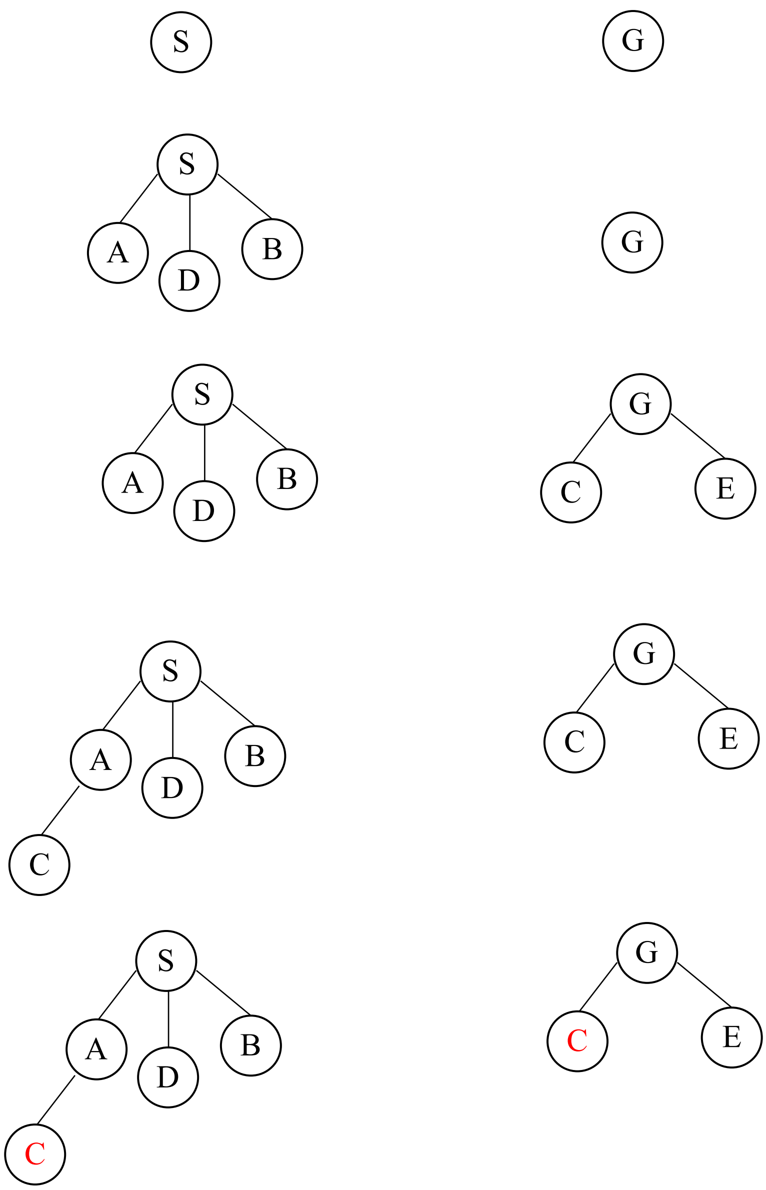


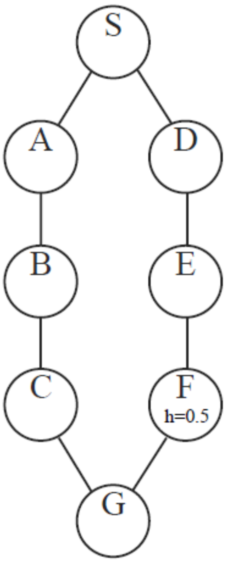
10
S
14
12
A
D
B
14
14
18
S
C
14
12
A
G
20
13<14 → f(G)=14



و) جستجوی دوطرفه را با استفاده از الگوریتم BFS و به صورت غیرهم‌زمان انجام دهید.

پاسخ)





۶- گراف مقابل را که در آن هزینه همه یال‌ها برابر با یک است، در نظر بگیرید. فرض کنید در حال طراحی هیوریستیکی برای آن هستید و تا کنون تنها h(F)=0.5 تعیین شده است و هیچ اطلاعات دیگری در دسترس ندارید.

الف) بازه‌ای از مقادیر را برای h(D) تعیین کنید که به ازای آن، این هیوریستیک قابل قبول و سازگار باشد.

پاسخ)

برای قابل‌قبول بودن، نباید هزینه مسیر نود D تا هدف، بیشتر از مقدار واقعی تخمین زده شود. هزینه واقعی رسیدن از نود D به هدف برابر با ۳ است، پس برای قابل قبول بودن باید $0 \leq h(D) \leq 3$ باشد.

برای سازگار بودن باید روابط زیر برقرار باشد:

$$\mathbf{1)}\ h(D) \leq c(D,a,E) + h(E) \ \rightarrow \ h(D) \leq 1 + h(E)$$

$$\mathbf{2)}\ h(E) \leq c(E,a,F) + h(F) \ \rightarrow \ h(E) \leq 1 + h(F) \rightarrow h(E) \leq 1.5$$

به طرفین عبارت ۲، یک واحد اضافه می‌کنیم:

$$\mathbf{3)}\ \rightarrow 1 + h(E) \leq 1 + 1.5$$

با روابط ۱ و ۳ به نتیجه روبرو می‌رسیم: $h(D) \leq 1 + 1.5 \ \rightarrow \ h(D) \leq 2.5$

پس برای سازگاری و قابل‌قبول بودن باید رابطه زیر برقرار باشد:

$$0 \leq h(D) \leq 2.5$$

ب) اگر h(E)=1.1 و مقادیر هیوریستیک تمامی گره‌های دیگر به جز B برابر با صفر در نظر گرفته شوند، بازه مقادیر h(B) را به‌گونه‌ای تعیین کنید که در حین اجرای جستجوی گرافی A* ترتیب بسط نودها به‌صورت S، A، D، E، B و F باشد.

پاسخ)

برای این که گره E زودتر از گره B گسترش یابد، باید رابطه زیر برقرار باشد:

$$f(E) \leq f(B) \rightarrow 2 + h(E) \leq 2 + h(B) \rightarrow 2 + 1.1 \leq 2 + h(B) \rightarrow 1.1 \leq h(B)$$

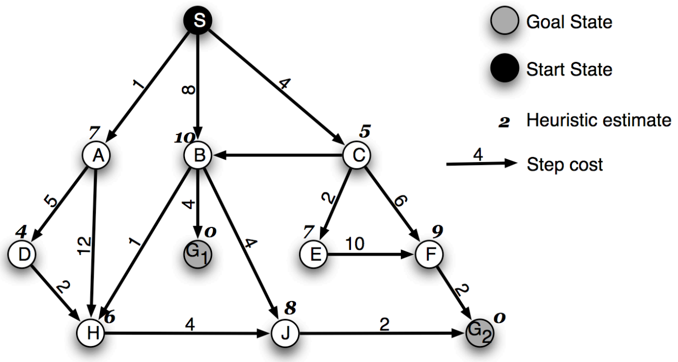
برای این که گره B زودتر از گره F گسترش یابد، باید رابطه زیر برقرار باشد:

$$f(B) \leq f(F) \rightarrow 2 + h(B) \leq 3 + h(F) \rightarrow h(B) \leq 1 + h(F) \rightarrow h(B) \leq 1.5$$

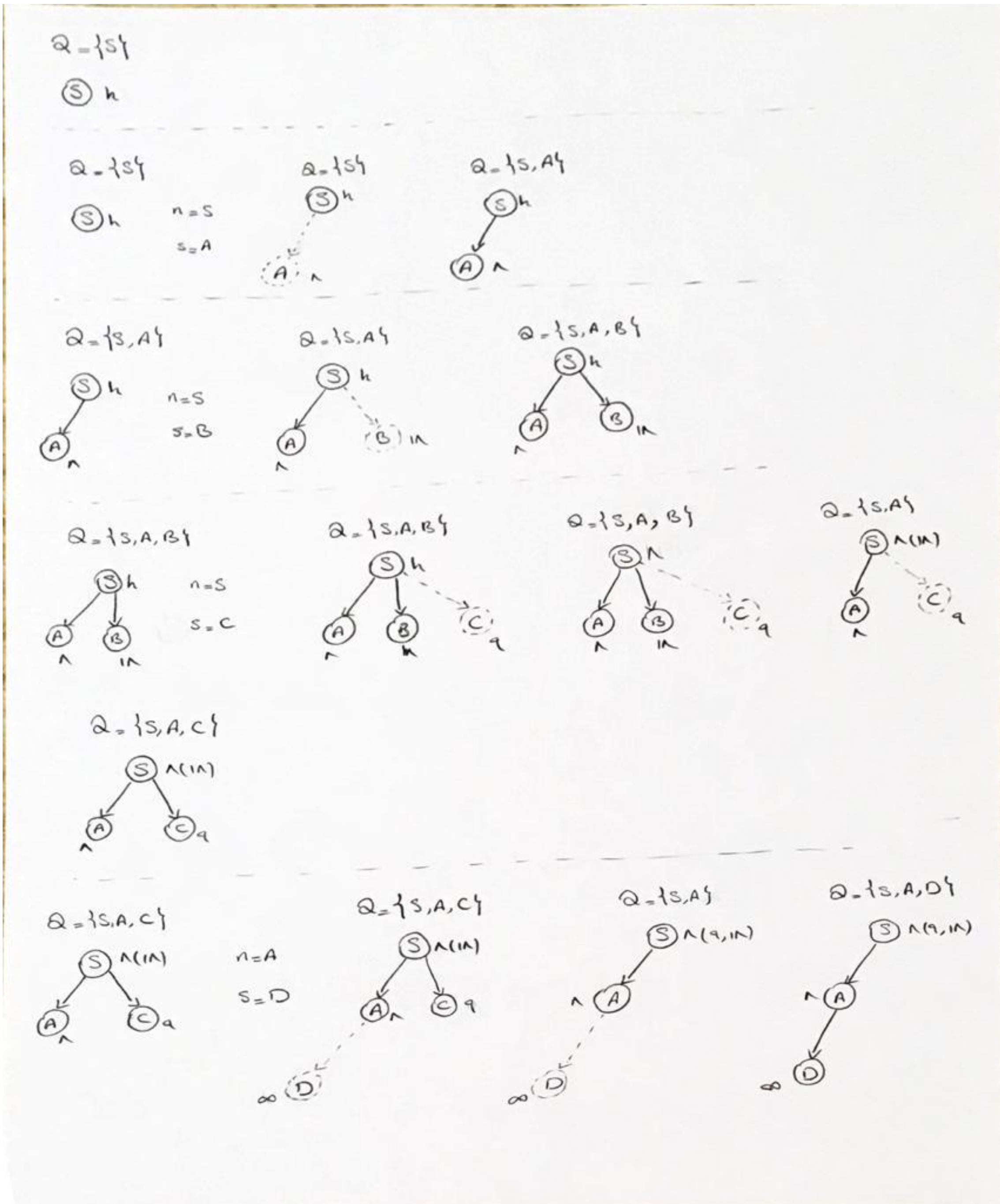
با اجتماع ۲ رابطه بالا به رابطه زیر می‌رسیم:

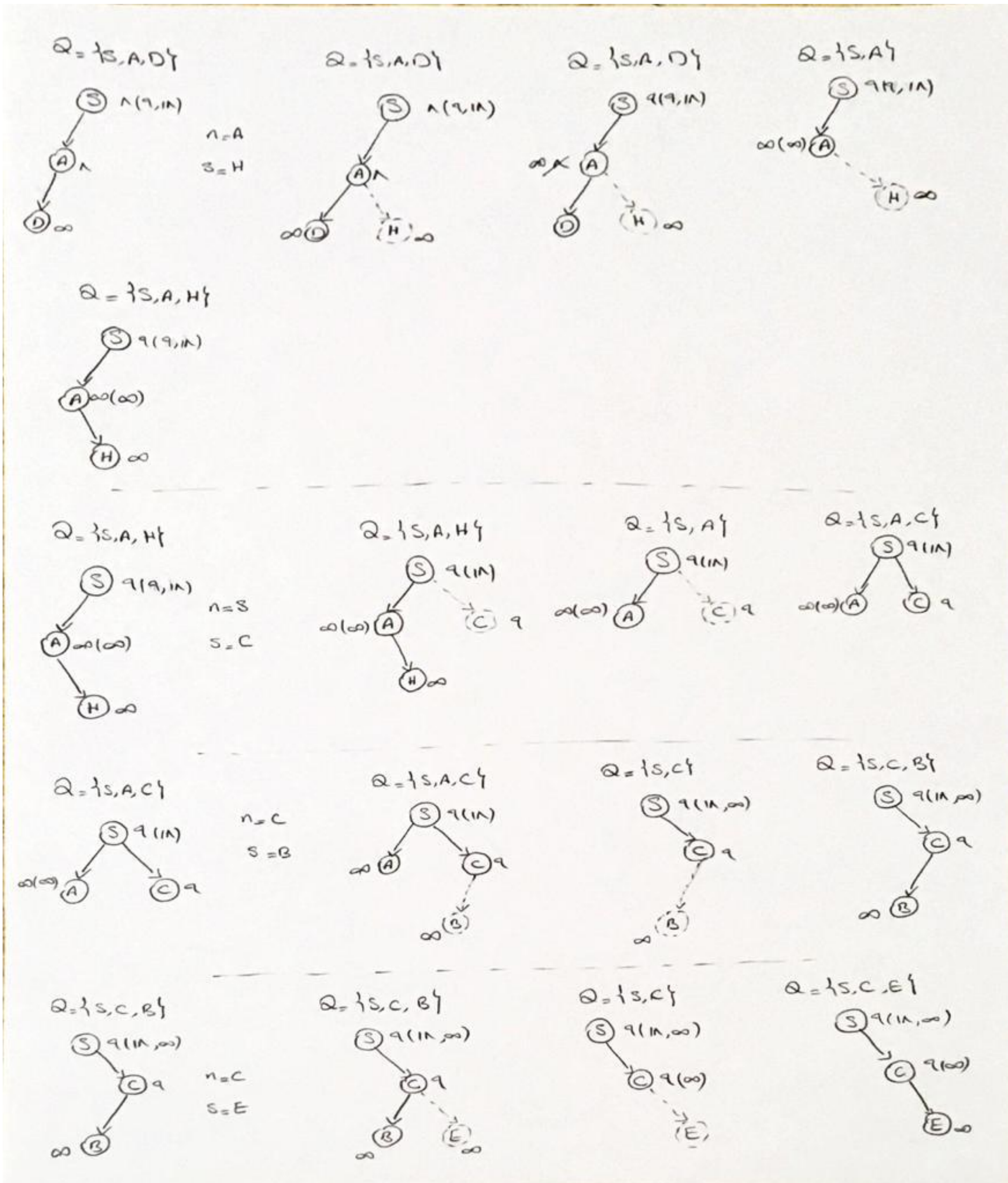
$$1.1 \ \leq h(B) \leq 1.5$$

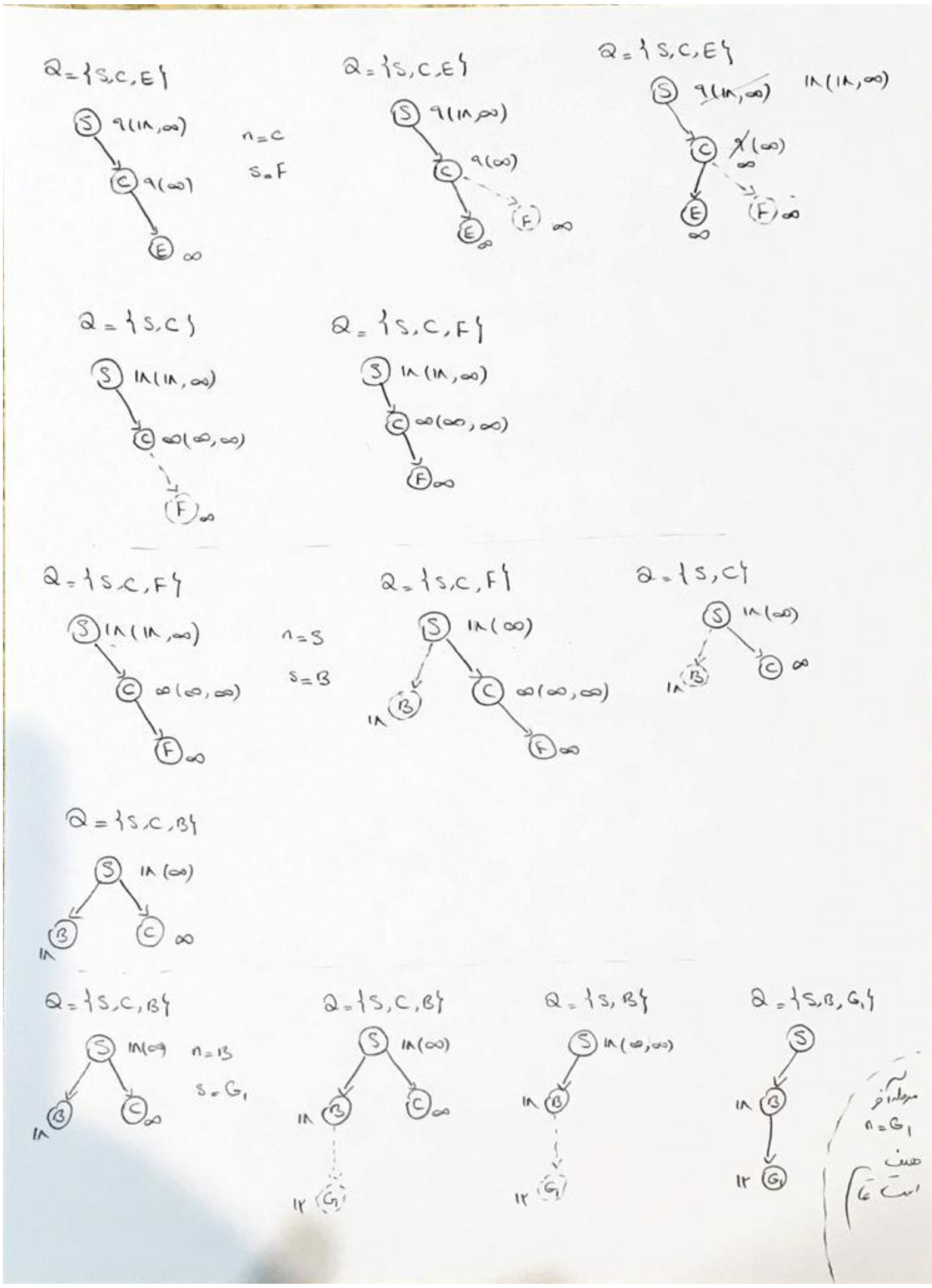
۷- الگوریتم SMA* را با در نظر گرفتن تنها سه خانه حافظه بر روی گراف زیر اجرا کنید.



پاسخ)




Q = {S, A, D}
n = A
s = H
Q = {S, A, D}
Q = {S, A, D}
Q = {S, A}
Q = {S, A, H}
Q = {S, A, H}
n = S
s = C
Q = {S, A, H}
Q = {S, A}
Q = {S, A, C}
Q = {S, A, C}
n = C
s = B
Q = {S, A, C}
Q = {S, C}
Q = {S, C, B}
Q = {S, C, B}
n = C
s = E
Q = {S, C, B}
Q = {S, C}
Q = {S, C, E}


Q = {S, C, E}
S ۹(۱۸, ∞)
C ۹(∞)
E ∞
n = C
s = F
Q = {S, C, E}
S ۹(۱۸, ∞)
C ۹(∞)
E ∞
F ∞
Q = {S, C, E}
S ۹(۱۸, ∞)
۱۸(۱۸, ∞)
C ۹(∞)
E ∞
F ∞
Q = {S, C}
S ۱۸(۱۸, ∞)
C ∞(∞, ∞)
F ∞
Q = {S, C, F}
S ۱۸(۱۸, ∞)
C ∞(∞, ∞)
F ∞
Q = {S, C, F}
S ۱۸(۱۸, ∞)
C ∞(∞, ∞)
F ∞
n = S
s = B
Q = {S, C, F}
S ۱۸(∞)
B ۱۸
C ∞(∞, ∞)
F ∞
Q = {S, C}
S ۱۸(∞)
B ۱۸
C ∞
Q = {S, C, B}
S ۱۸(∞)
B ۱۸
C ∞
Q = {S, C, B}
S ۱۸(∞)
B ۱۸
C ∞
n = B
s = G1
Q = {S, C, B}
S ۱۸(∞)
B ۱۸
C ∞
G1 ۱۲
Q = {S, B}
S ۱۸(∞, ∞)
B ۱۸
G1 ۱۲
Q = {S, B, G1}
S
B ۱۸
G1 ۱۲
n = G1