



دانشگاه صنعتی امیرکبیر



دانشکده مهندسی کامپیوتر

مبانی و کاربردهای هوش مصنوعی

جلسه اول کلاس حل تمرین

مهر ۱۴۰۰

محیط وظیفه و انواع محیط ها و فرموله بندی مسائل



QA) مشخص کردن PEAS در عامل‌های **دستیار صوتی اپل** و **پیشنهاد دهنده دوست** و مشخص کردن ویژگی‌های محیط وظیفه عامل‌های فوق که شامل موارد زیر هستند:

- ❖ Fully observable / partially observable
- ❖ Single agent / multi-agent
- ❖ Deterministic / stochastic
- ❖ Discrete / continuous
- ❖ Episodic / sequential
- ❖ Static / dynamic
- ❖ Known / unknown

حسگرها	عملگرها	محیط	معیار کارایی	
میکروفون جهت تشخیص صوت	نمایش نتایج بر روی صفحه، تولید صدا، اجرای ابزارها و ...	سیستم عاملی که برنامه بر روی آن اجرا میشود	انجام درست درخواست کاربران و میزان رضایت از آن	دستیار صوتی اپل
اطلاعات ورودی کاربر	نمایش به کاربر، ایمیل به کاربر	هر نوع شبکه اجتماعی	درصد پیشنهاد هایی که به دوستی منجر شده است	پیشنهاد دهنده‌ی دوست

شناخته / ناشناخته	گسسته / پیوسته	مرحله ای / ترتیبی	ایستا / پویا	قطعی / تصادفی	تک عاملی/چند عاملی	قابل مشاهده	
شناخته	پیوسته	ترتیبی	پویا	تصادفی	تک عاملی	جزئی	دستیار صوتی اپل
شناخته	گسسته	مرحله ای	پویا	تصادفی	تک عاملی	جزئی	پیشنهاد دهنده‌ی دوست

QA عامل خودروی بدون سرنشین

یک خودروی بدون سرنشین در محیطی مطابق روبرو قرار دارد و اعمال مجاز عامل عبارتند از:

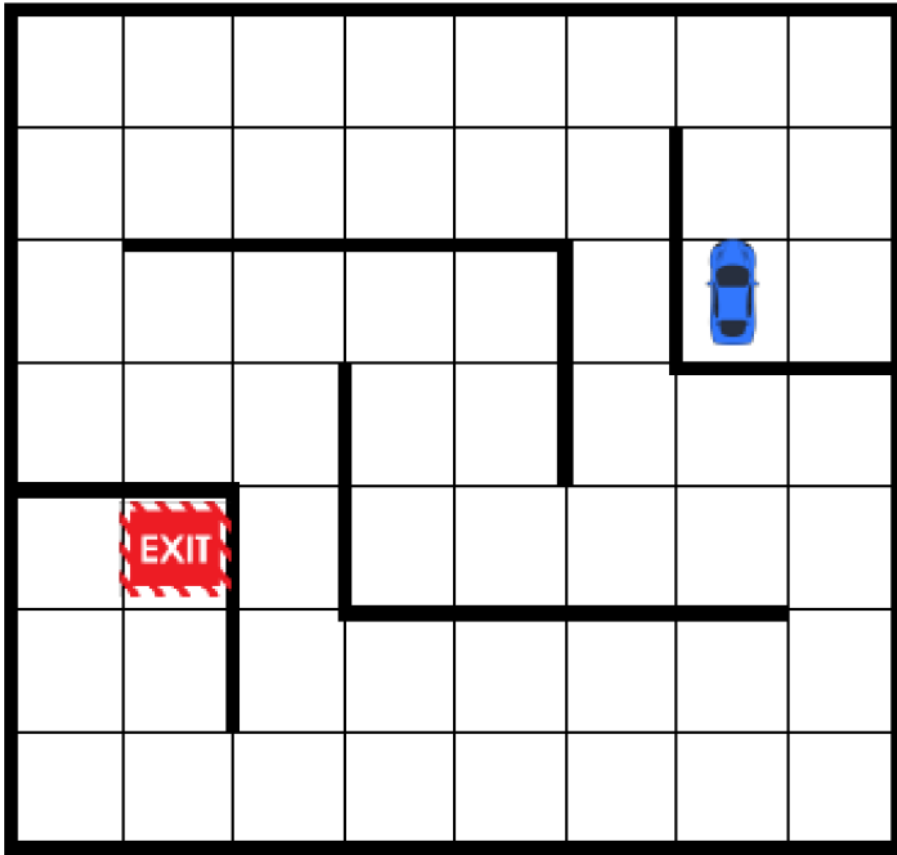
Right ❖

Left ❖

Fast ❖

Slow ❖

سرعت عامل تنها می تواند بین صفر و V_{max} باشد و چرخش نیز تنها در صورتی رخ میدهد که سرعت برابر صفر باشد.



الف) در صورتی که محیط عامل یک جدول $N \times M$ باشد با فرض اینکه تمامی حالات از ابتدا قابل دسترس باشند، اندازه کل فضای حالت چقدر است؟ توضیح دهید.

$$M \times N \times 4 \times (V_{max} + 1)$$

ب) در صورتی که اعمال غیر مجاز توسط تابع انتقال حذف شوند حداکثر ضریب انشعاب چقدر است؟

جستجو های ناآگاهانه



💡 دو نکته مهم !

❖ جستجوی گرافی در برابر جستجوی درختی

- ❖ در جستجوی گرافی از یک حافظه اضافی برای نگه داری گره‌های که مشاهده شده است، استفاده میشود.
- ❖ در حالیکه در جستجوی درختی ممکن است یک گره چند بار visit شود.

❖ زمان انجام آزمون هدف (Goal test)

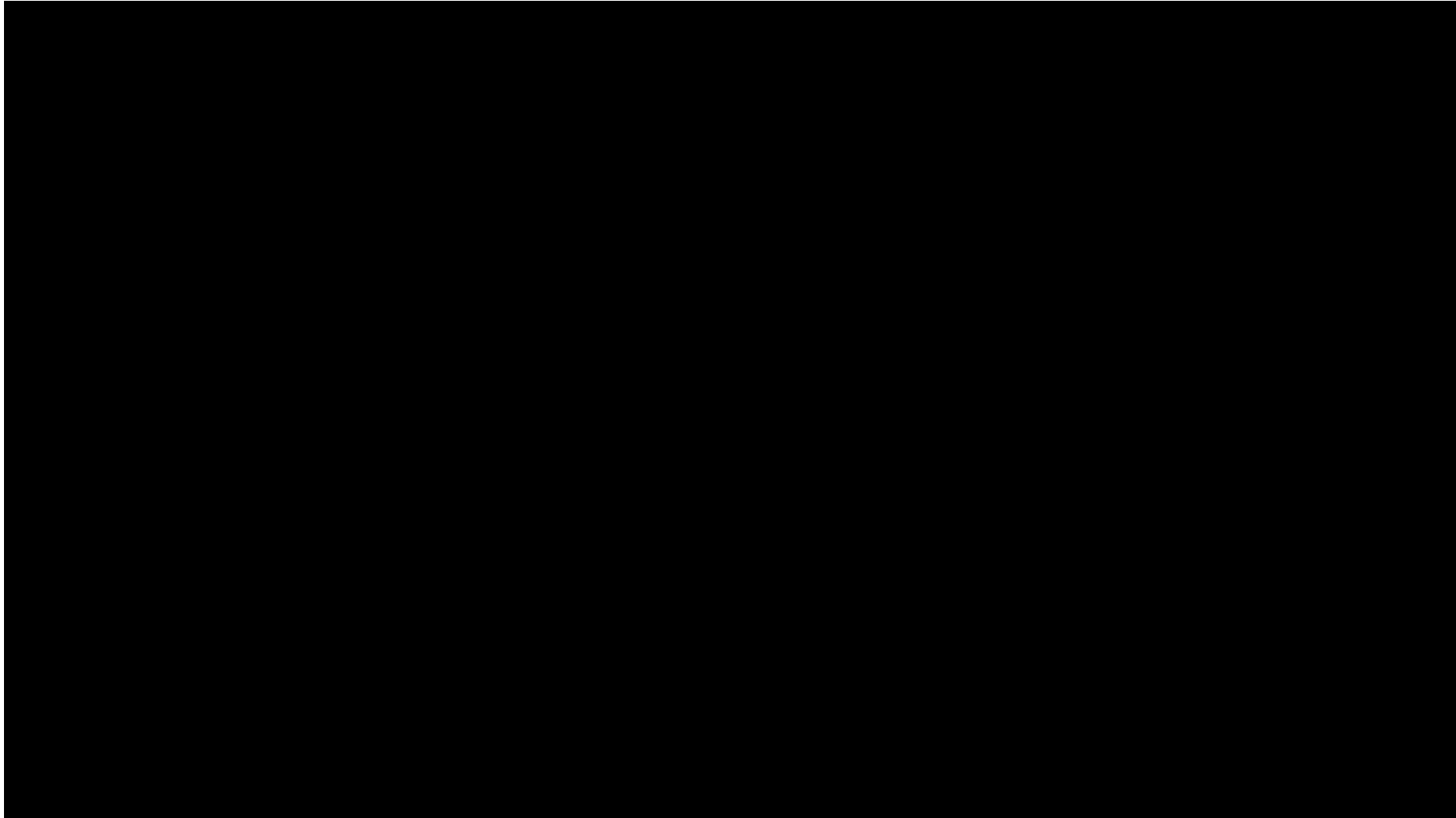
- ❖ در هنگام تولید نود
- ❖ در هنگام بسط نود

جستجوی اول سطح – BFS : بسط کم عمق ترین گره بسط نیافته

```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
frontier ← a FIFO queue with node as the only element
explored ← an empty set
loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier) /* chooses the shallowest node in frontier */
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
        child ← CHILD-NODE(problem, node, action)
        if child.STATE is not in explored or frontier then
            if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
            frontier ← INSERT(child, frontier)
```

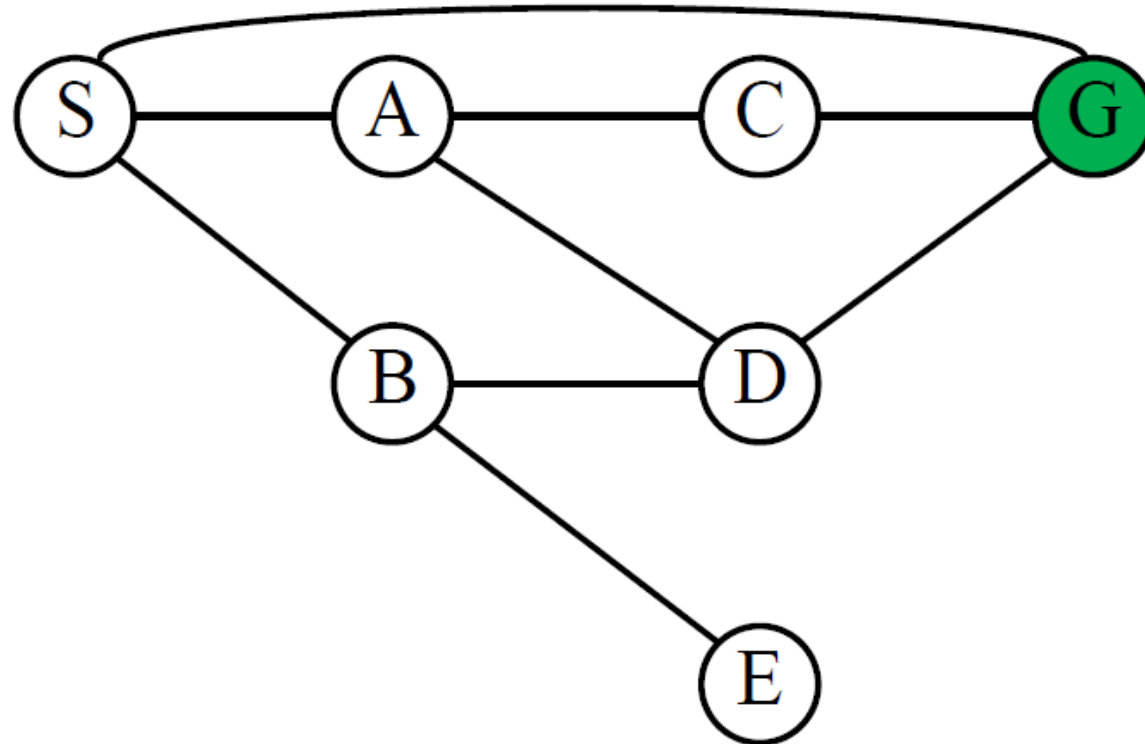
ویدیوی آموزش الگوریتم گرافی BFS همراه با مثال

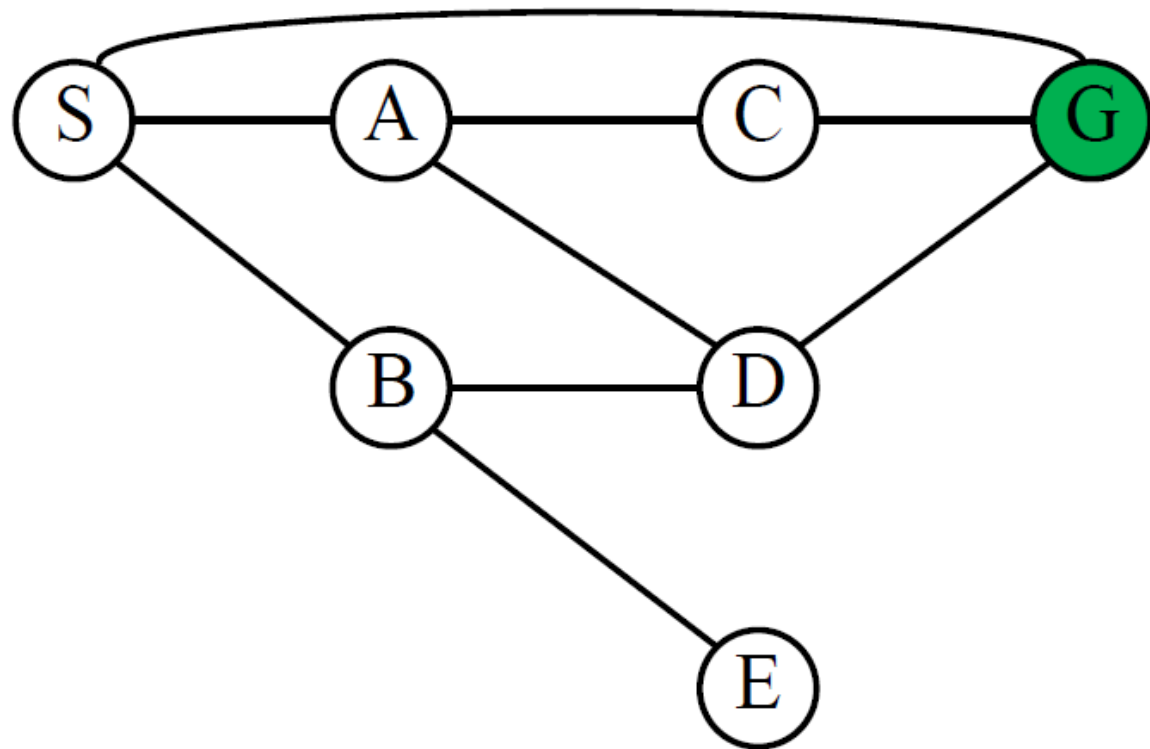
ویدیوی زیر در این [لینک](#) قابل مشاهده است.



QA جستجوی گرافی با استفاده از الگوریتم BFS

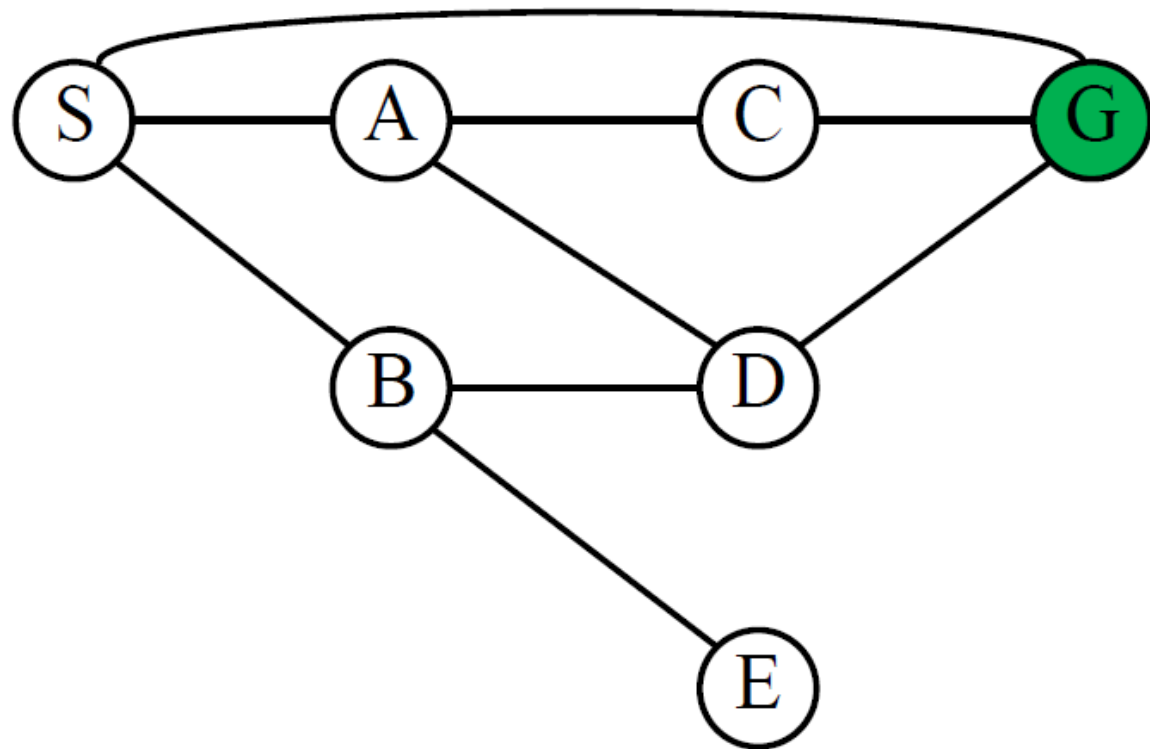
توجه : آزمون هدف در هنگام بسط نود انجام می پذیرد و در شرایط یکسان براساس حروف الفبا عمل شود.



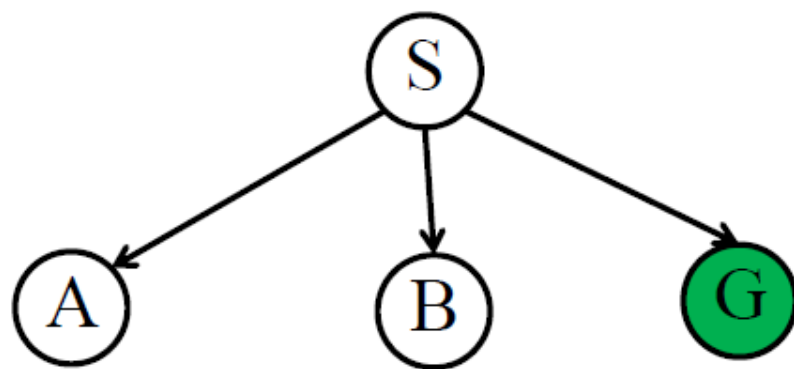


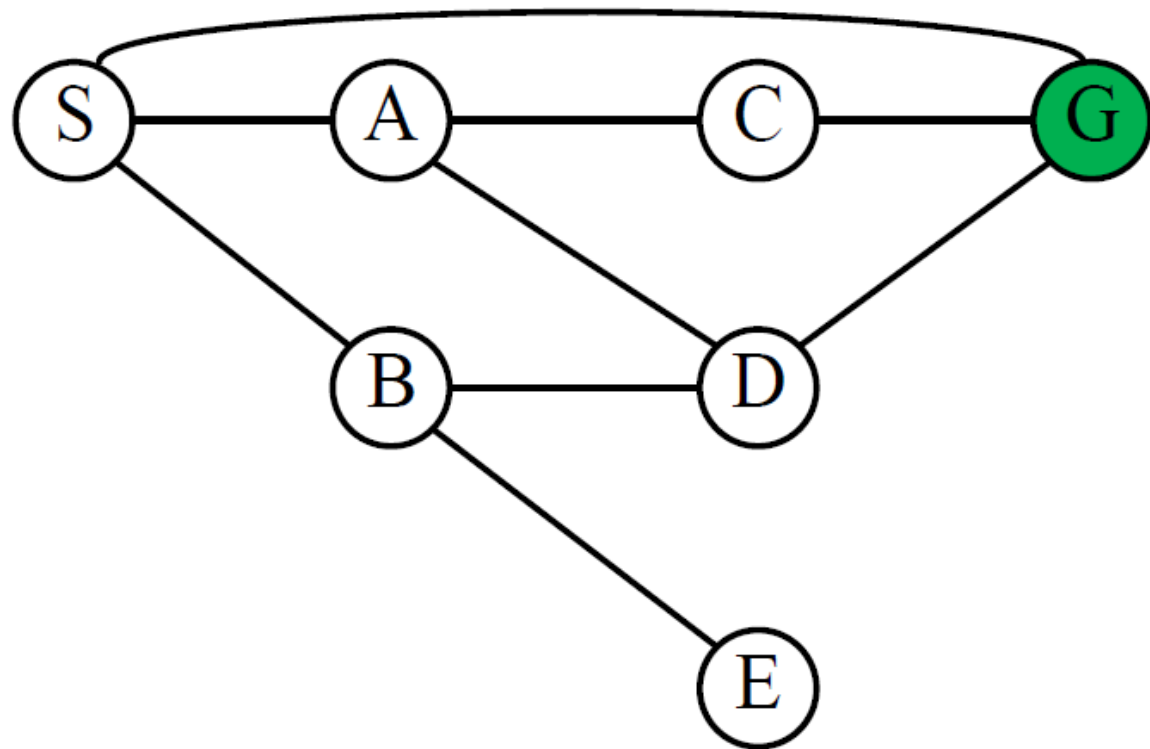
Frontier	Explored
S	



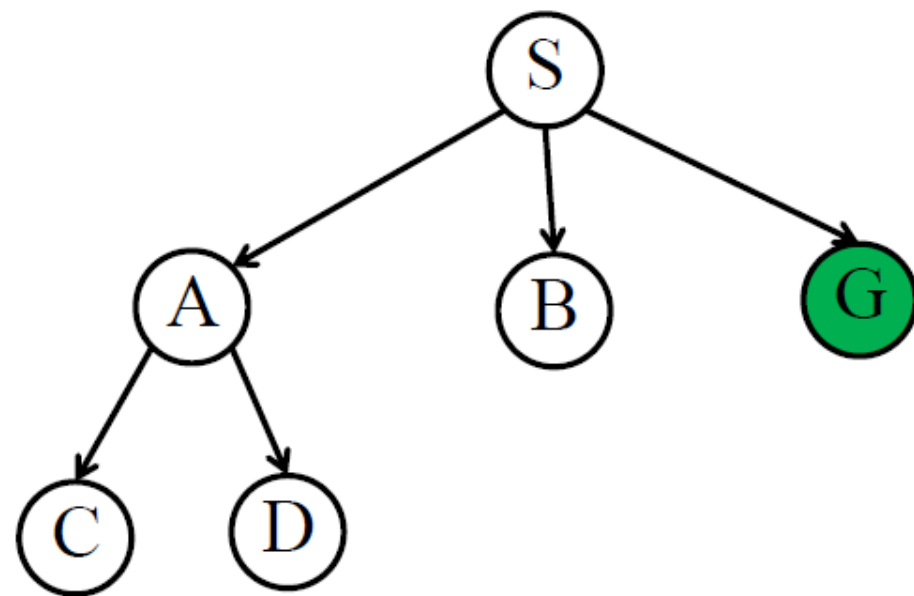


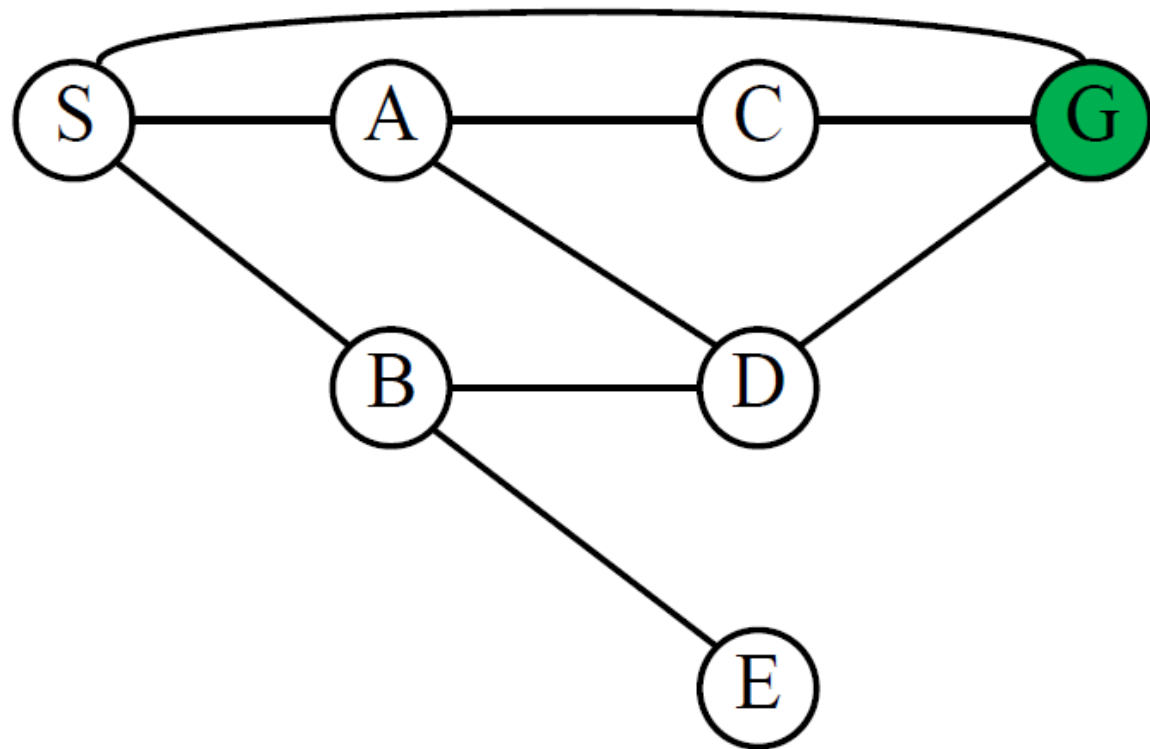
Frontier	Explored
S	
A, B, G	S



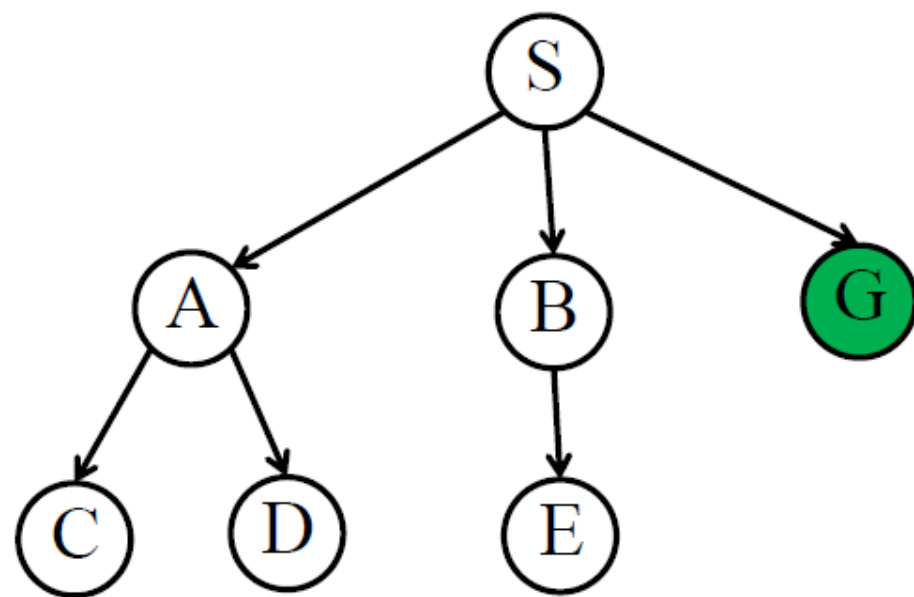


Frontier	Explored
S	
A, B, G	S
B, G, C, D	S, A



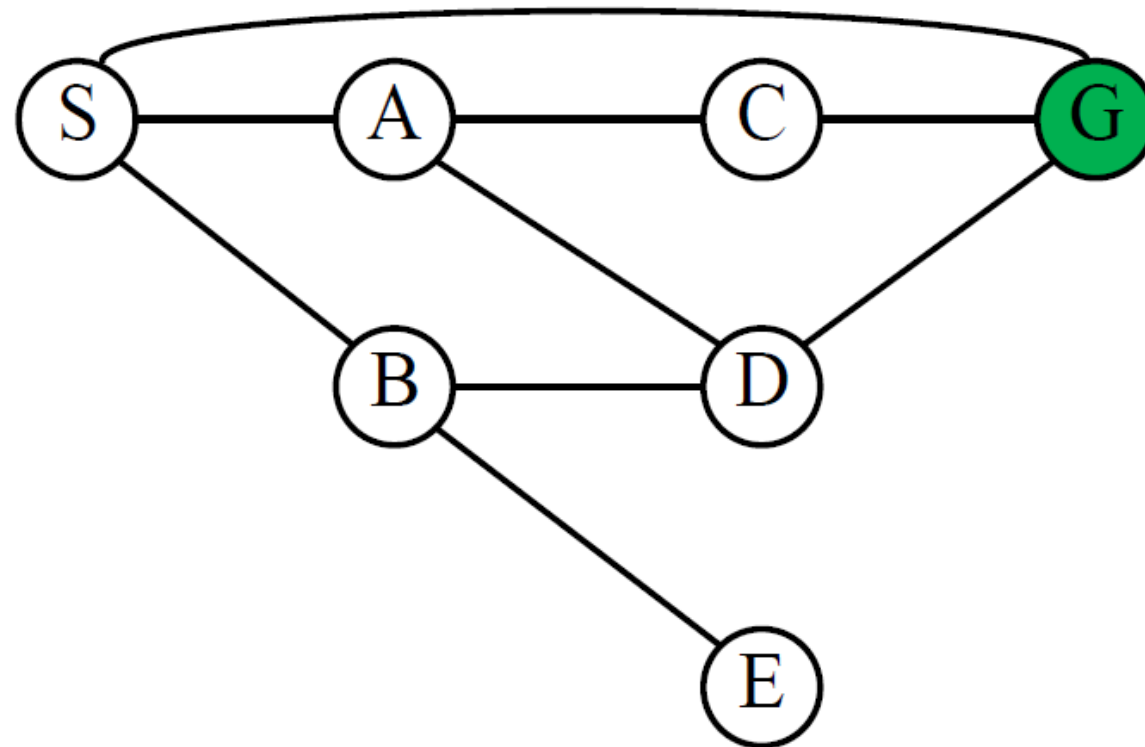


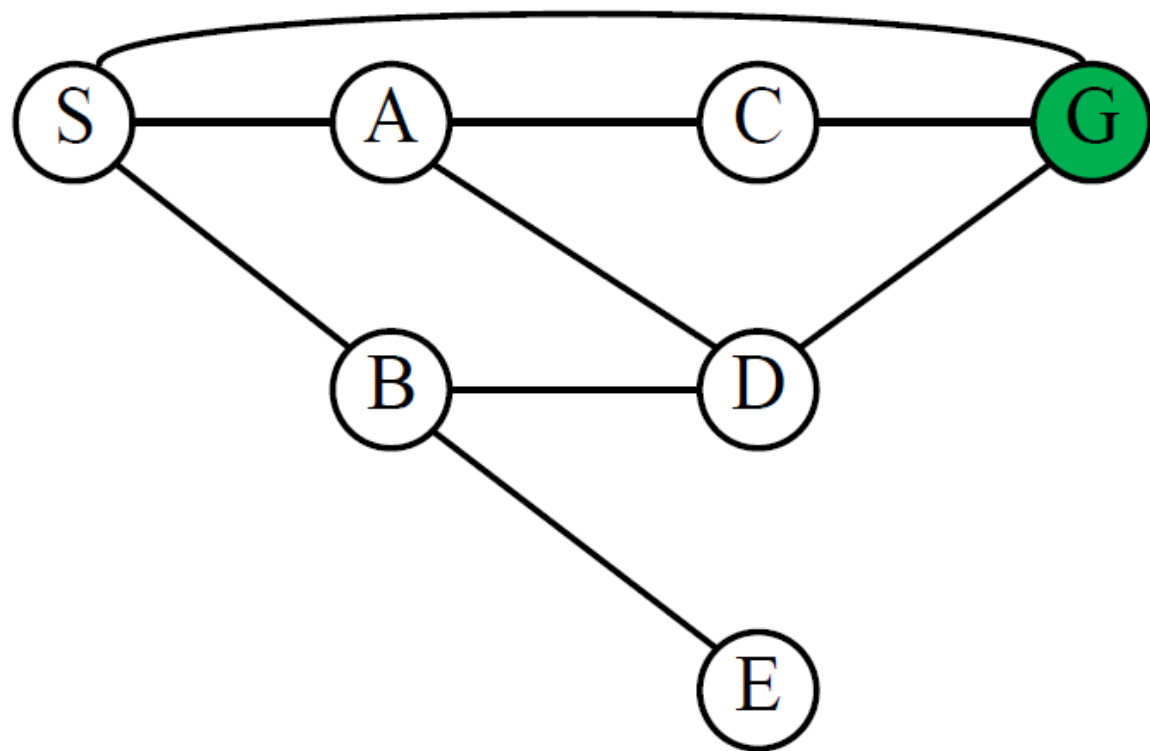
Frontier	Explored
S	
A, B, G	S
B, G, C, D	S, A
G, C, D, E	S, A, B



جستجوی درختی با استفاده از الگوریتم BFS

توجه : آزمون هدف در هنگام بسط نود انجام می پذیرد و در شرایط یکسان براساس حروف الفبا عمل شود.

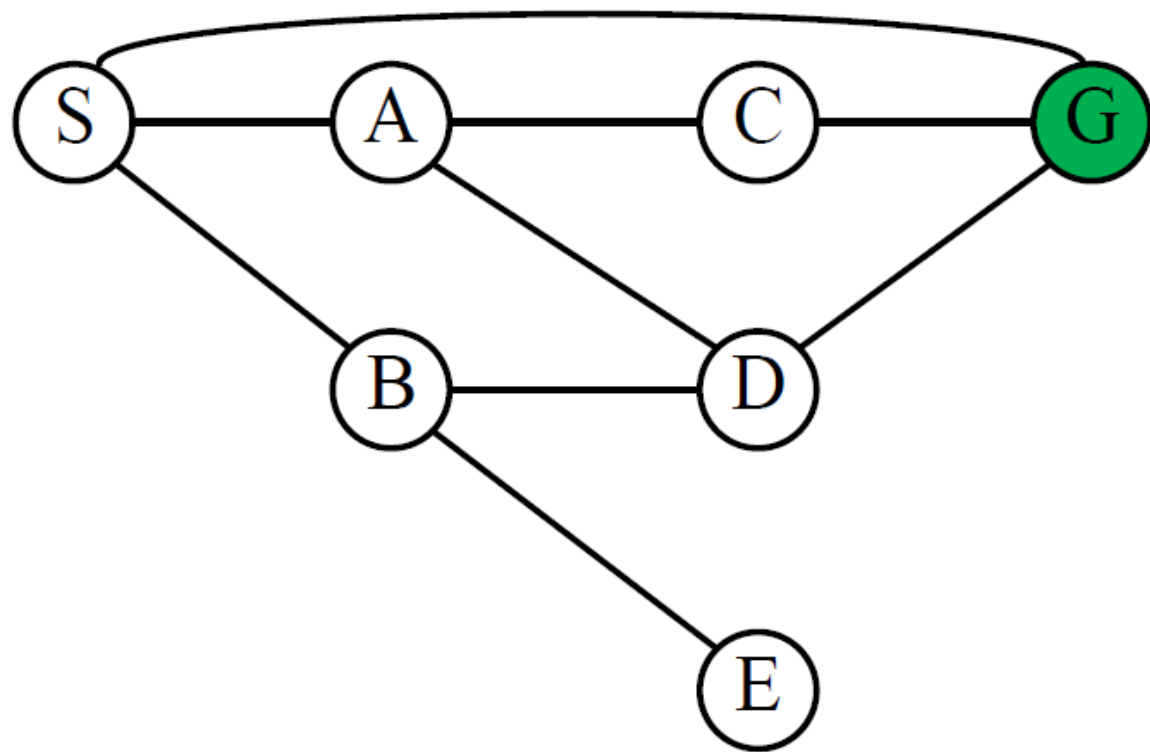




Frontier

S

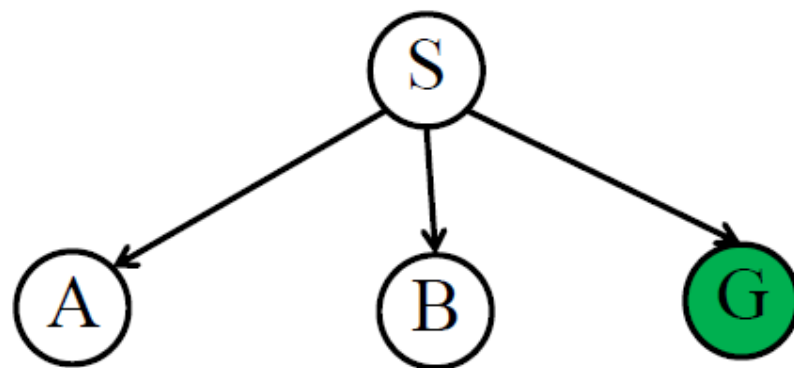
S

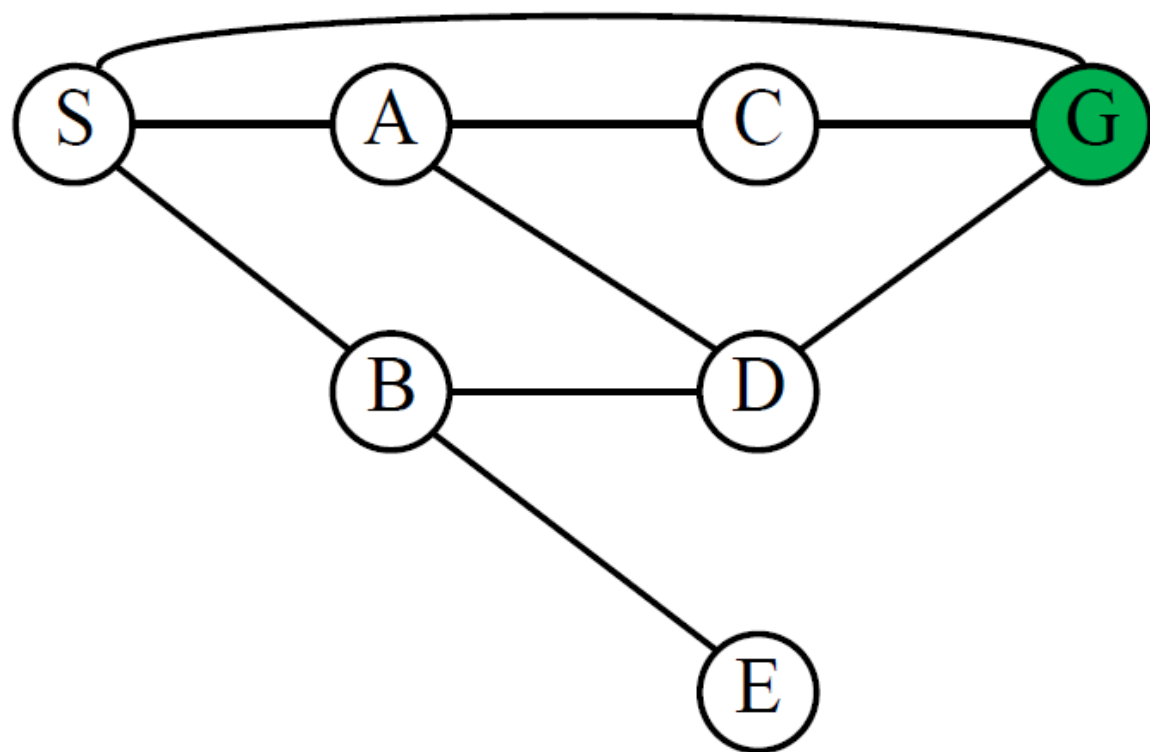


Frontier

S

A, B, G



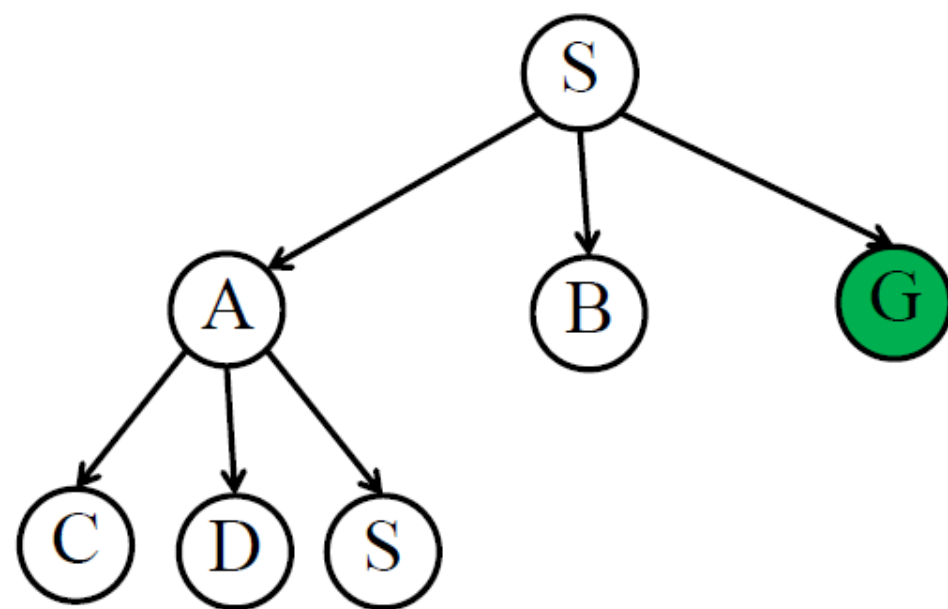


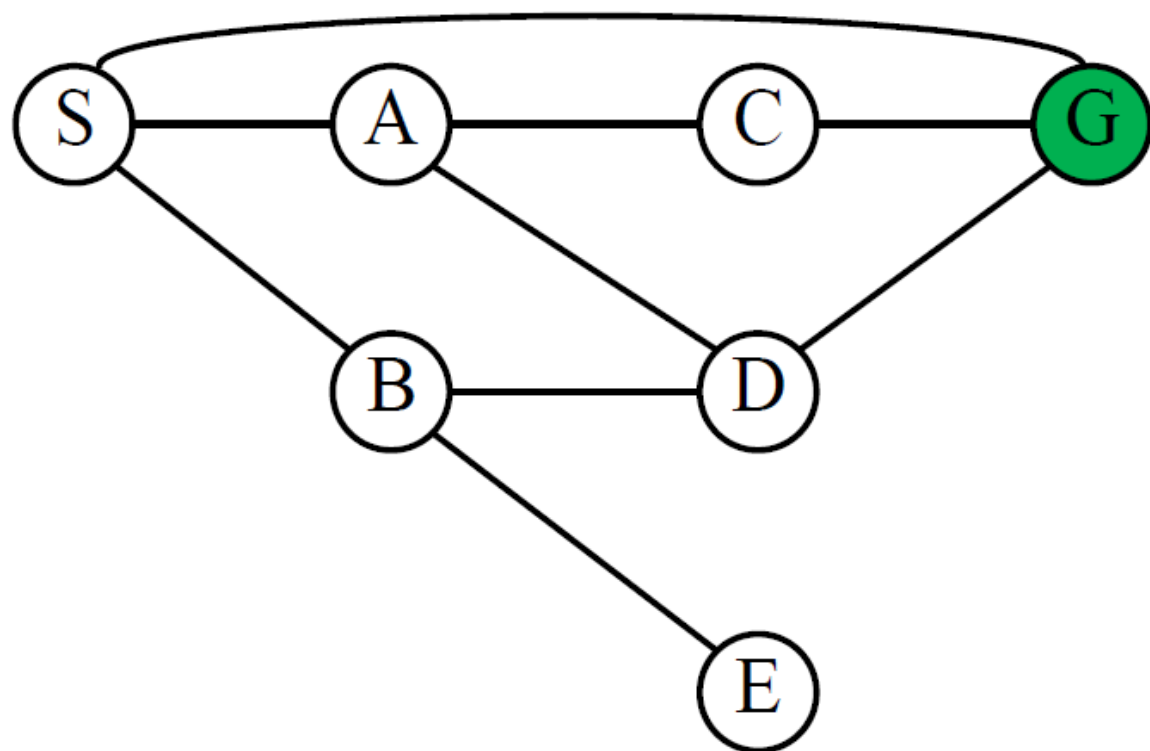
Frontier

S

A, B, G

B, G, C, D, S





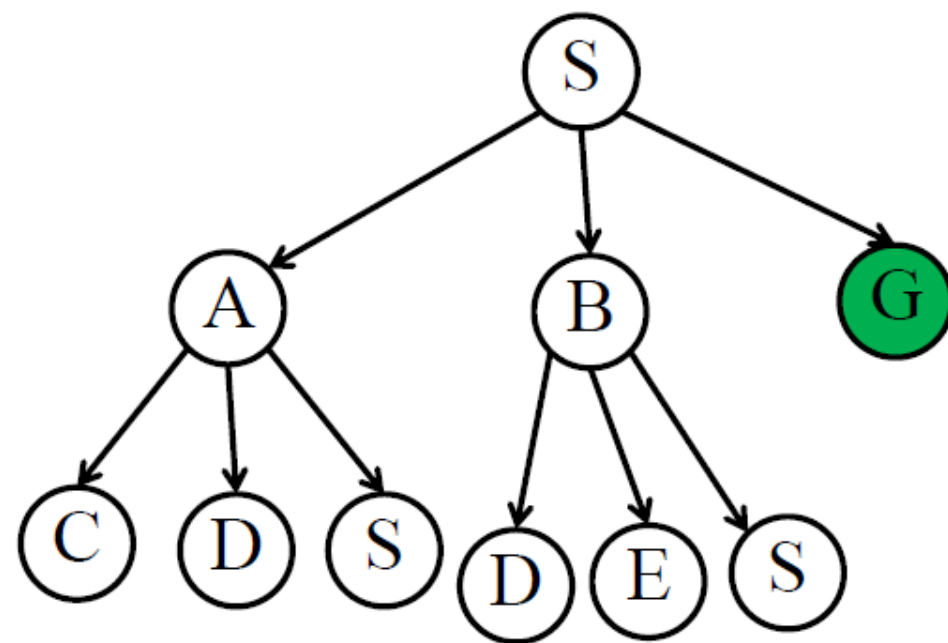
Frontier

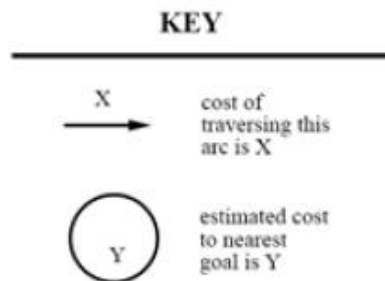
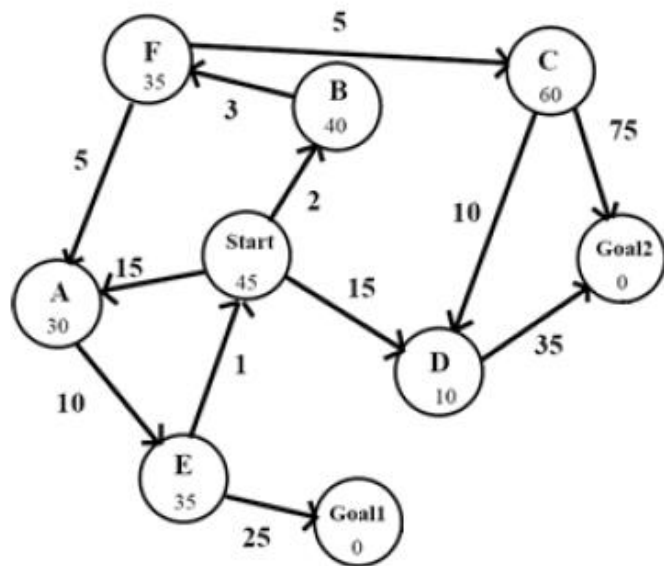
S

A, B, G

B, G, C, D, S

G, C, D, S, D, E, S

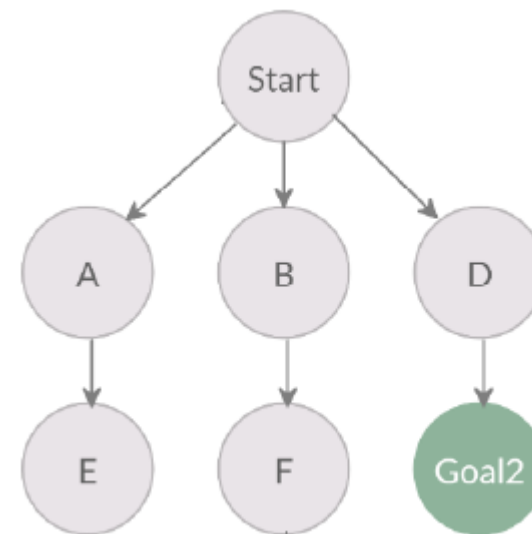




QA مثالی دیگر از BFS

آزمون هدف هنگام تولید نود صورت می پذیرد و جستجو درختی است.

1. Start
2. Start \rightarrow A
3. Start \rightarrow B
4. Start \rightarrow D
5. Start \rightarrow A \rightarrow E
6. Start \rightarrow B \rightarrow F
7. Start \rightarrow D \rightarrow Goal2



آیا مسیری که BFS به آن رسیده است، مسیر بهینه است؟

جستجوی هزینه یکنواخت - UCS

❖ استراتژی : در هر لحظه گره ای برای بسط دادن انتخاب می شود که هزینه رسیدن به آن گره تا آن لحظه مینیمم باشد.

❖ هم ارز با BFS اگر هزینه هر یال برابر با ۱ باشد.

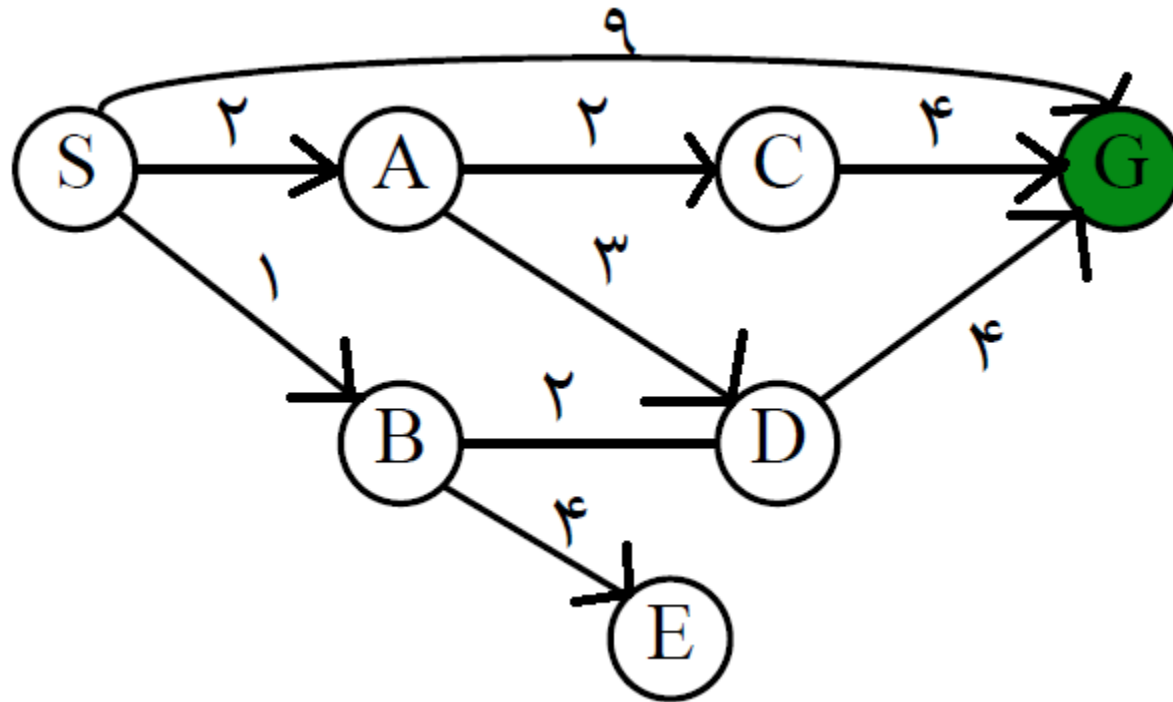
❖ دو تفاوت عمده با BFS

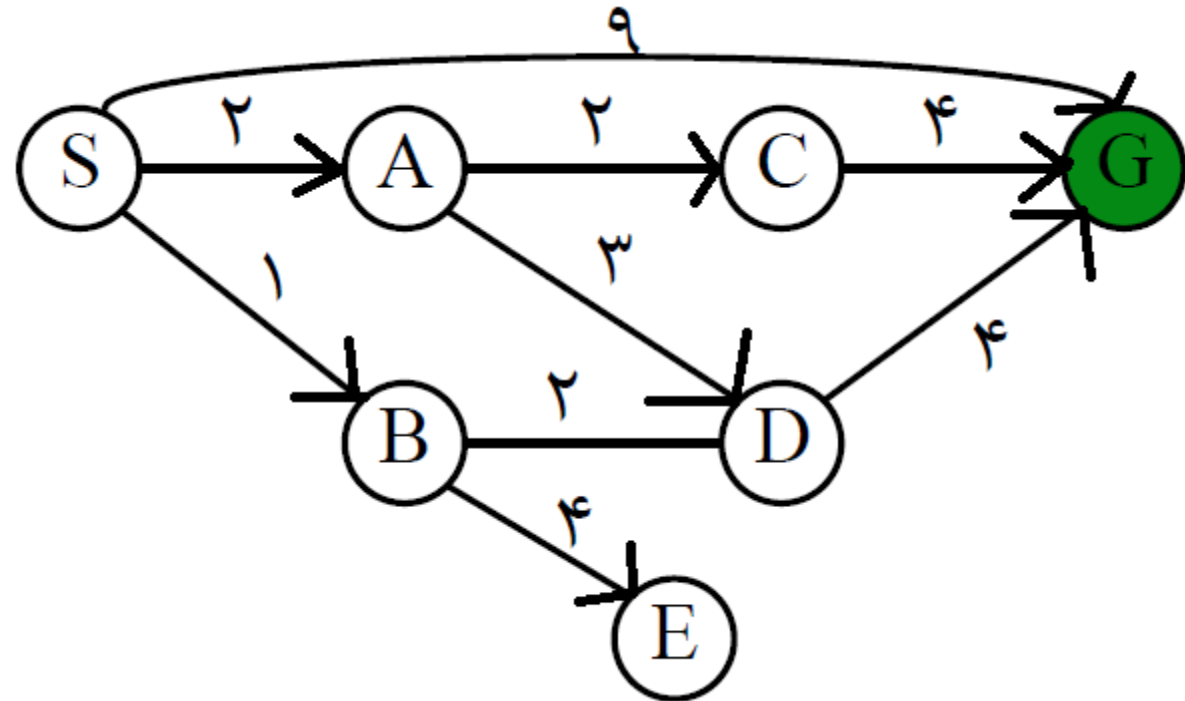
- آزمون هدف بر روی یک نود هنگام انتخاب آن برای بسط اعمال میشود.
- فرض کنید نودی تولید شده باشد که وضعیت آن نود قبلا در مجموعه **frontier** وارد شده باشد، دراین صورت اگر نود جدید حاوی مسیر بهتری نسبت به نود قبلی باشد، آن نود را با نود موجود در **frontier** جایگزین می کنیم.

شبهه کد الگوریتم UCS

```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
frontier ← a priority queue ordered by PATH-COST, with node as the only element
explored ← an empty set
loop do
    if EMPTY?(frontier) then return failure
    node ← POP(frontier) /* chooses the lowest-cost node in frontier */
    if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
    add node.STATE to explored
    for each action in problem.ACTIONS(node.STATE) do
        child ← CHILD-NODE(problem, node, action)
        if child.STATE is not in explored or frontier then
            frontier ← INSERT(child, frontier)
        else if child.STATE is in frontier with higher PATH-COST then
            replace that frontier node with child
```

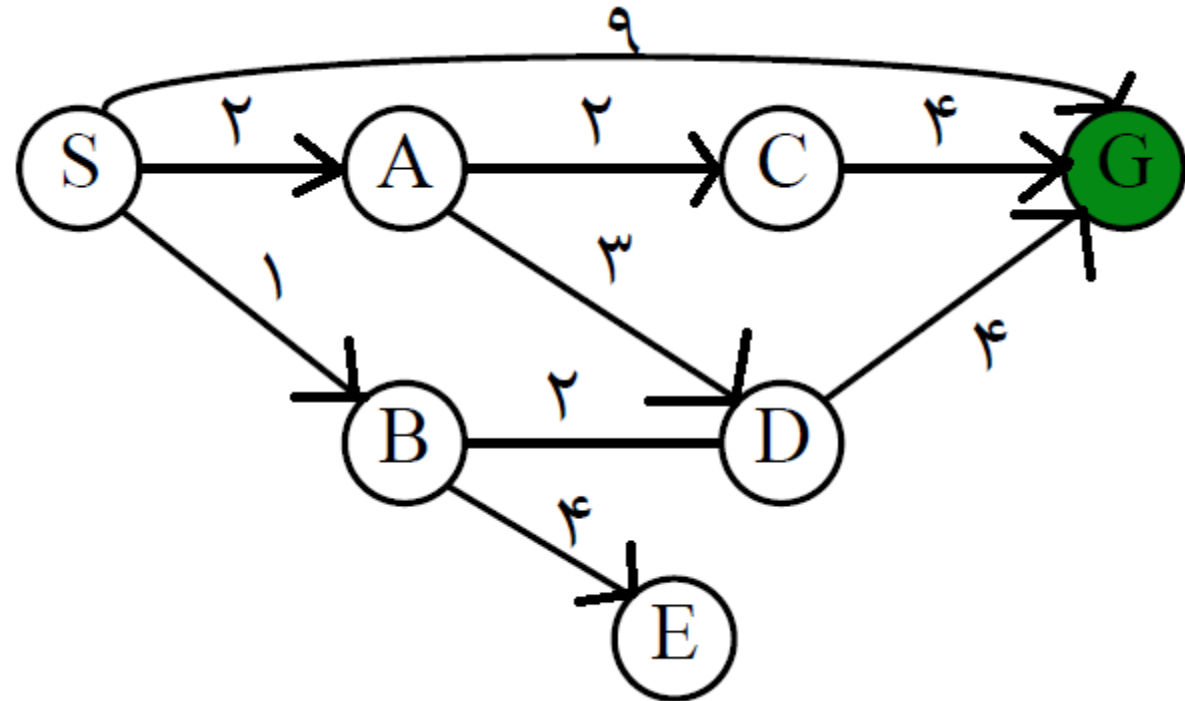
مثال جستجو با استفاده از الگوریتم UCS QA



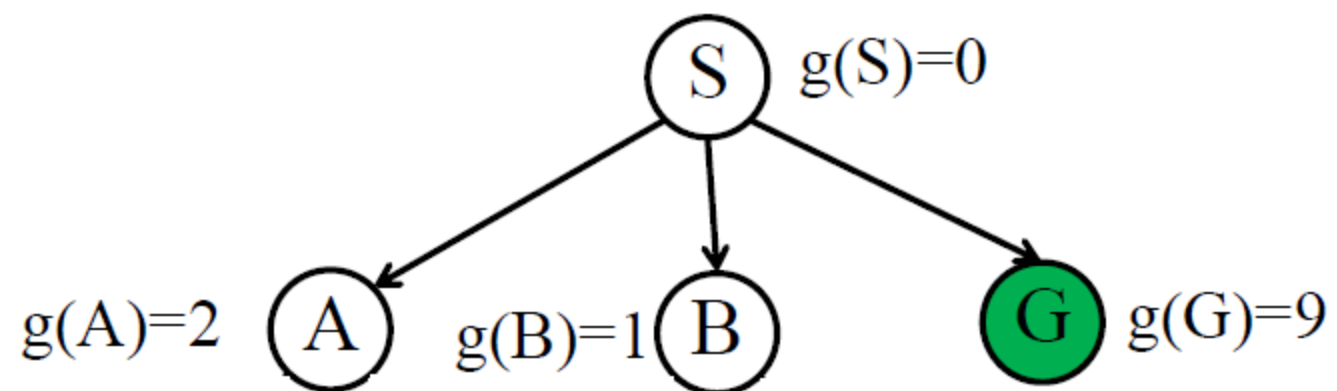


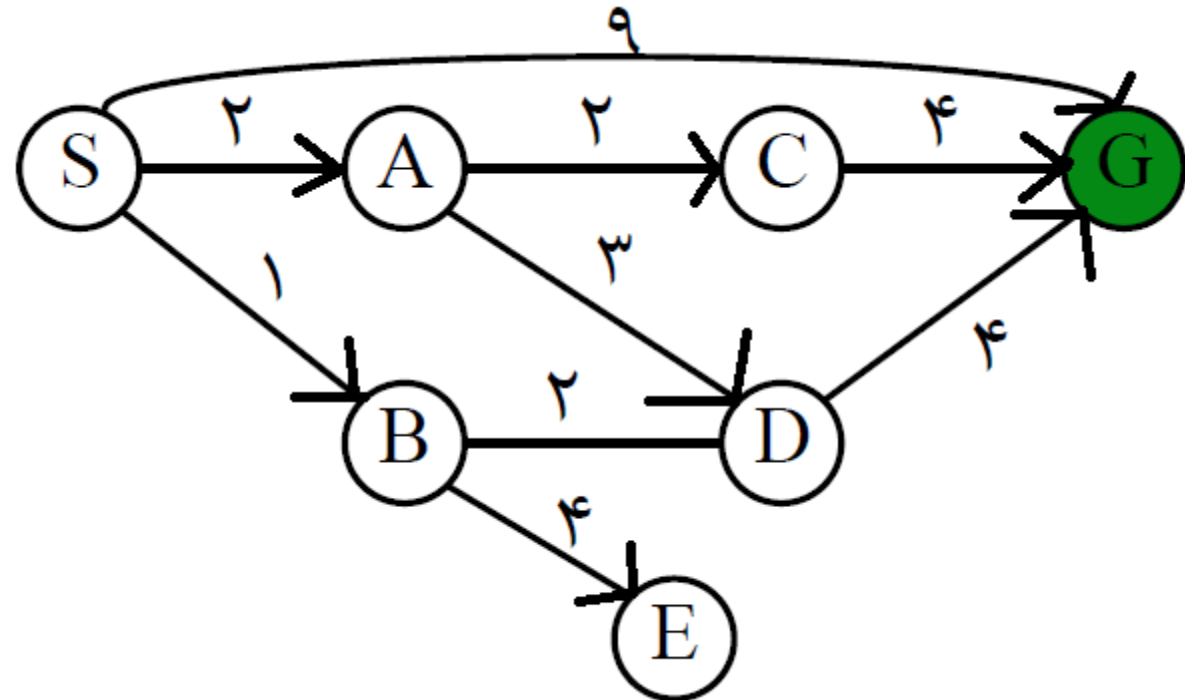
$$\textcircled{S} \quad g(S)=0$$

S

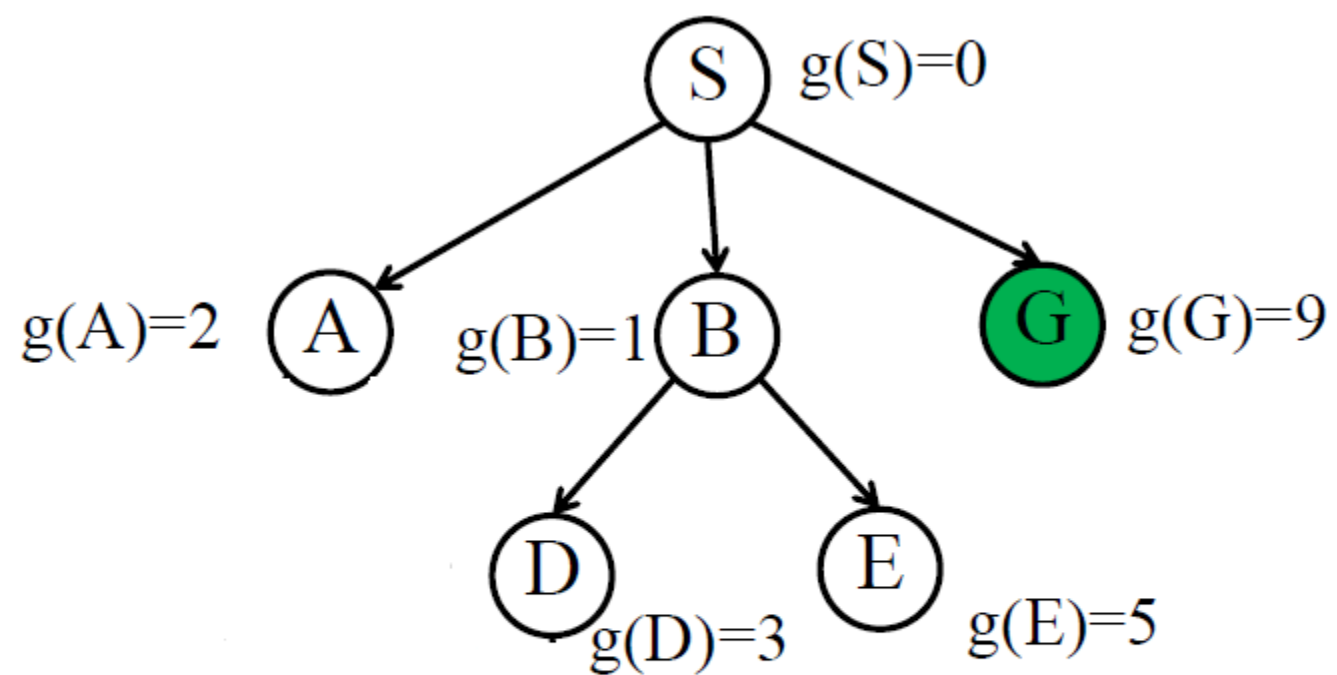


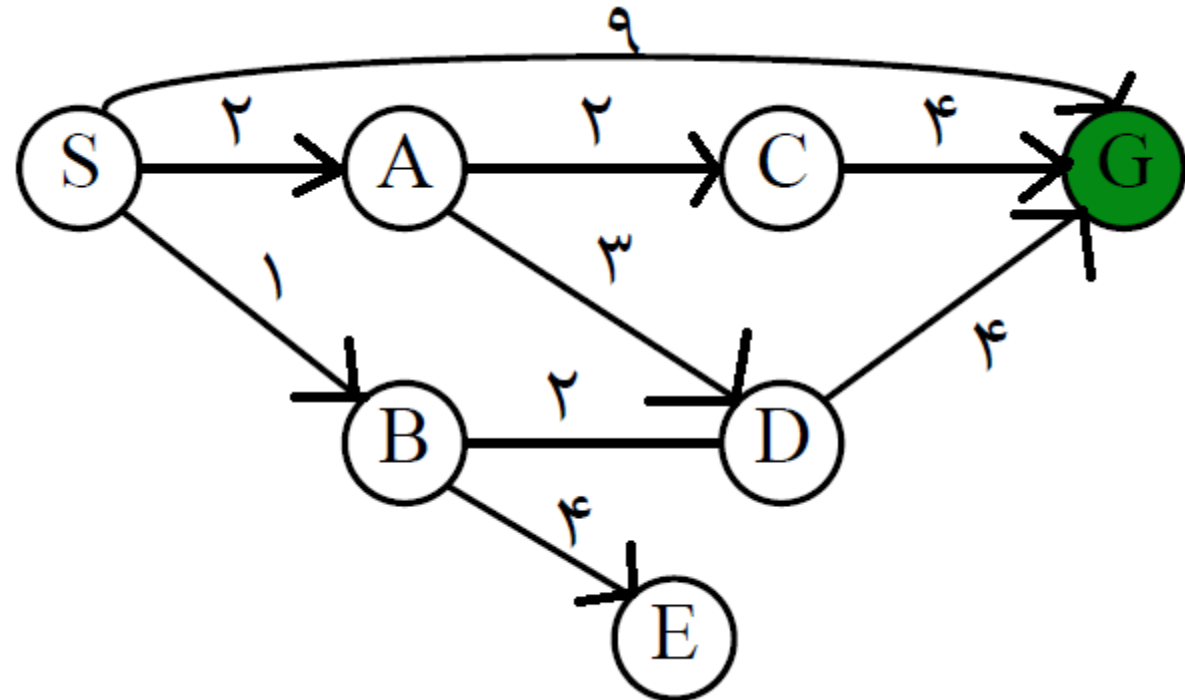
$S \rightarrow B$



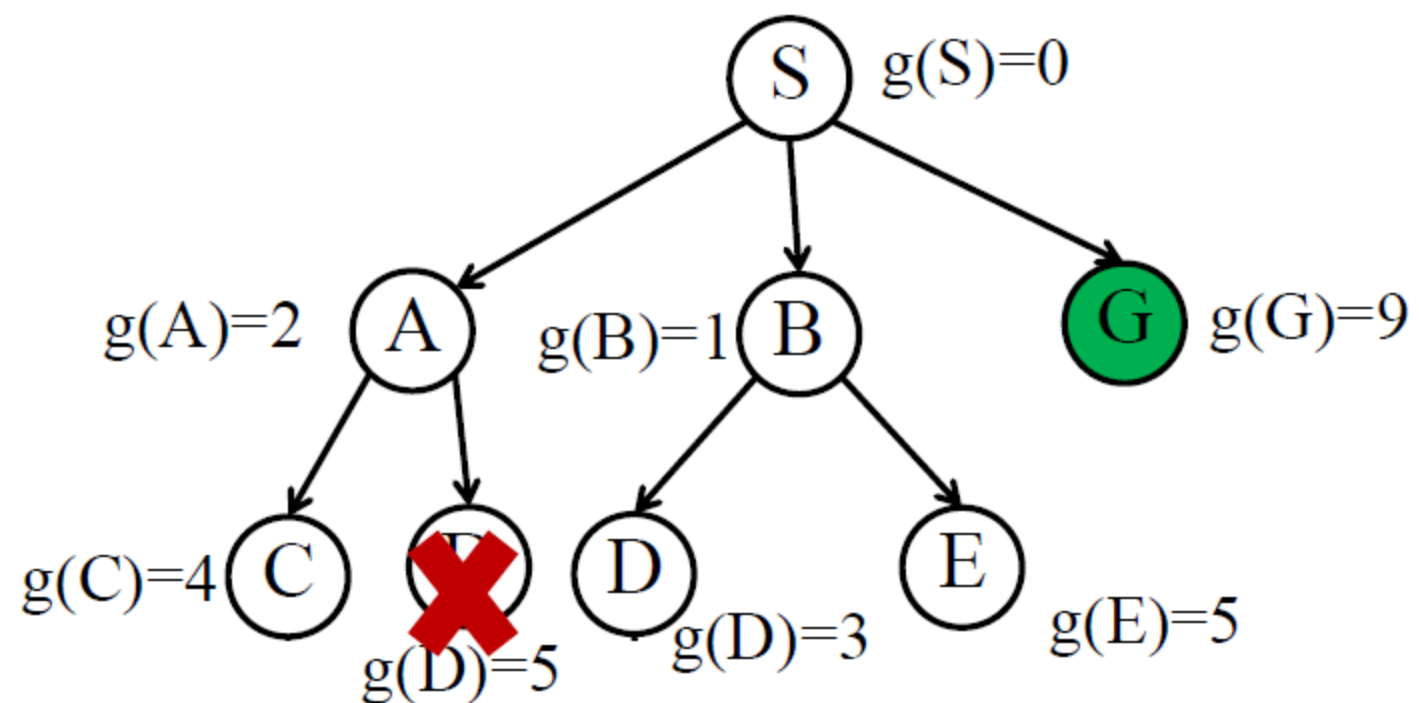


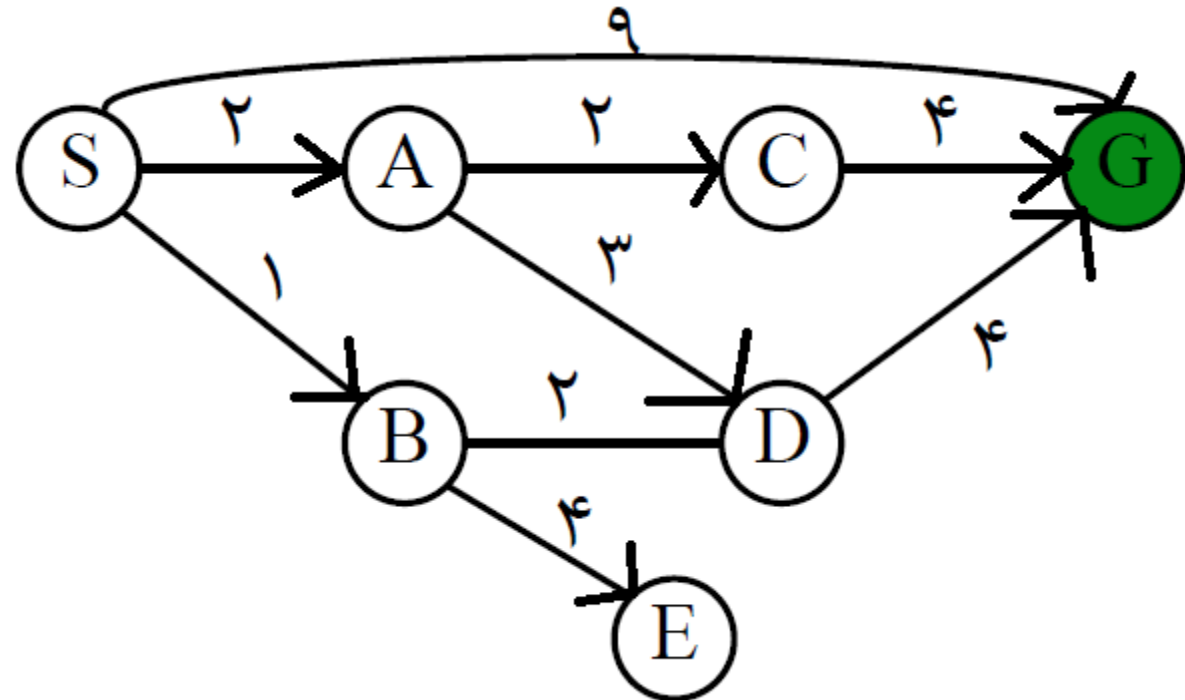
$S \rightarrow A$



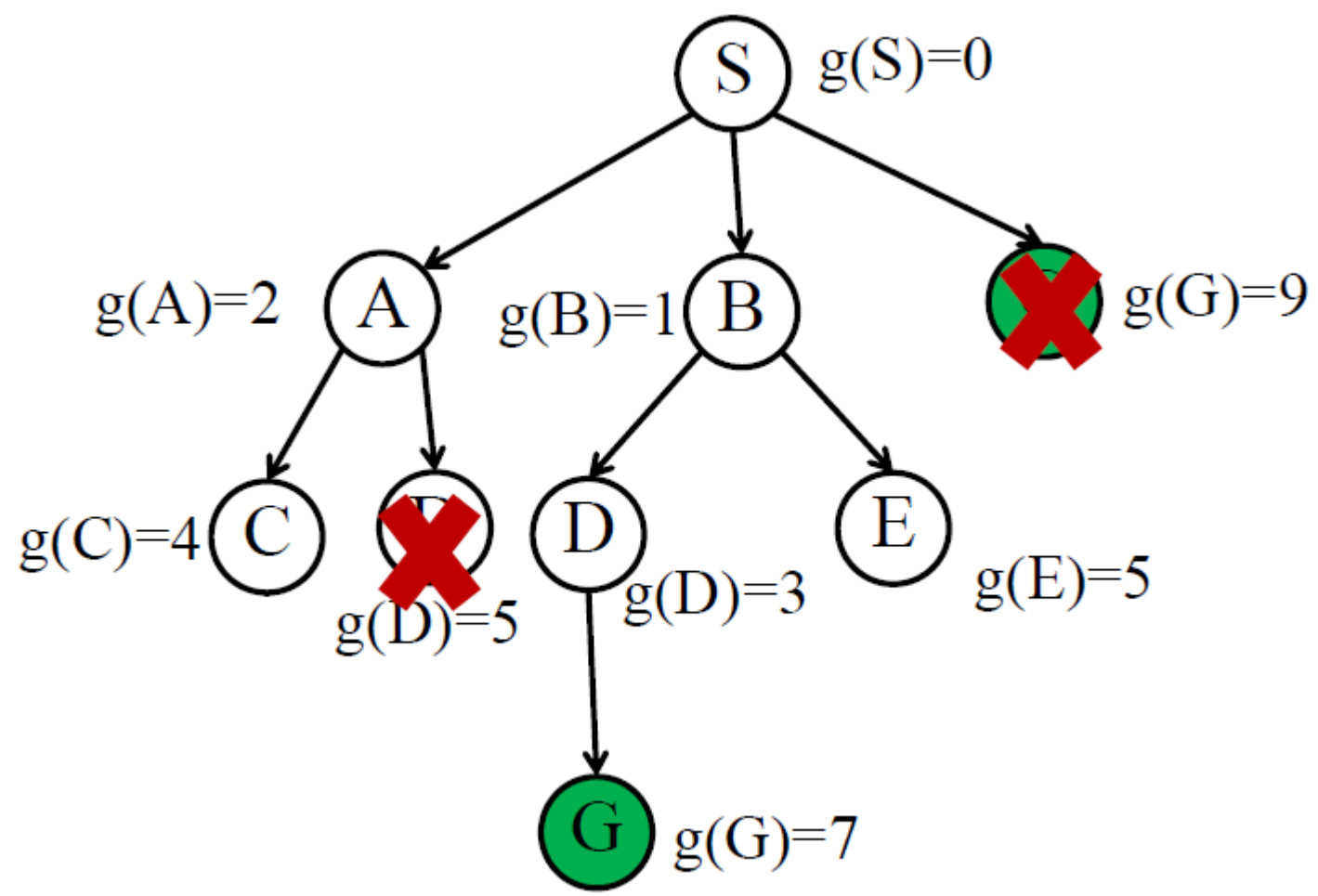


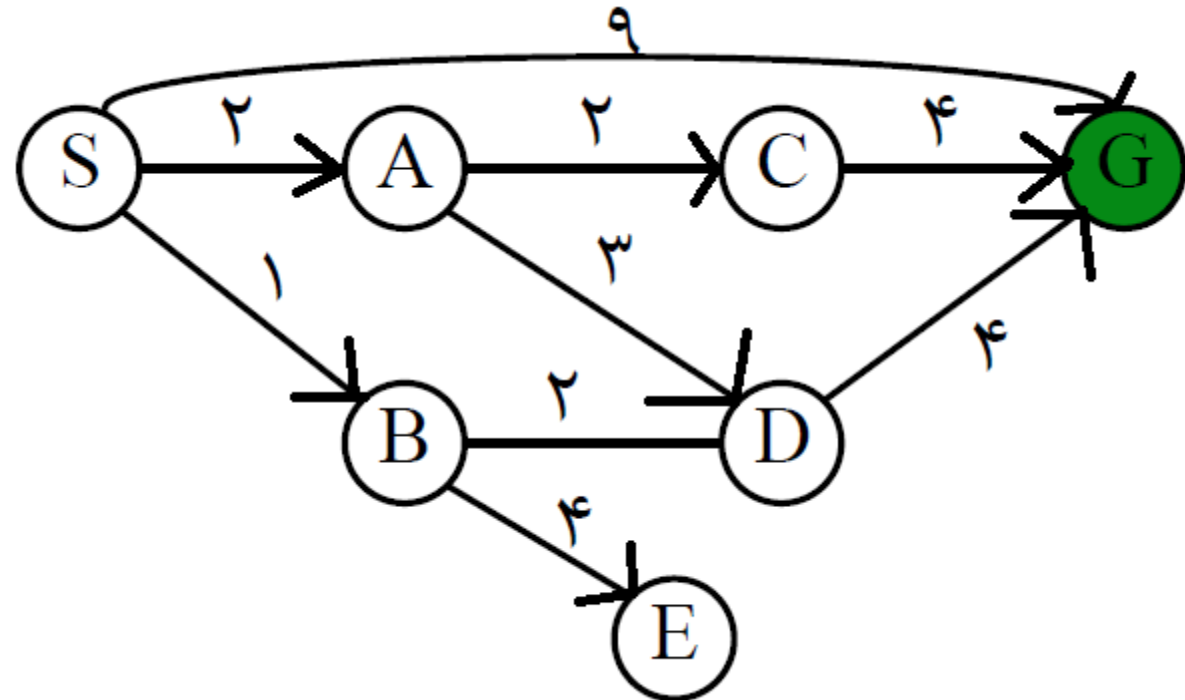
$S \rightarrow B \rightarrow D$



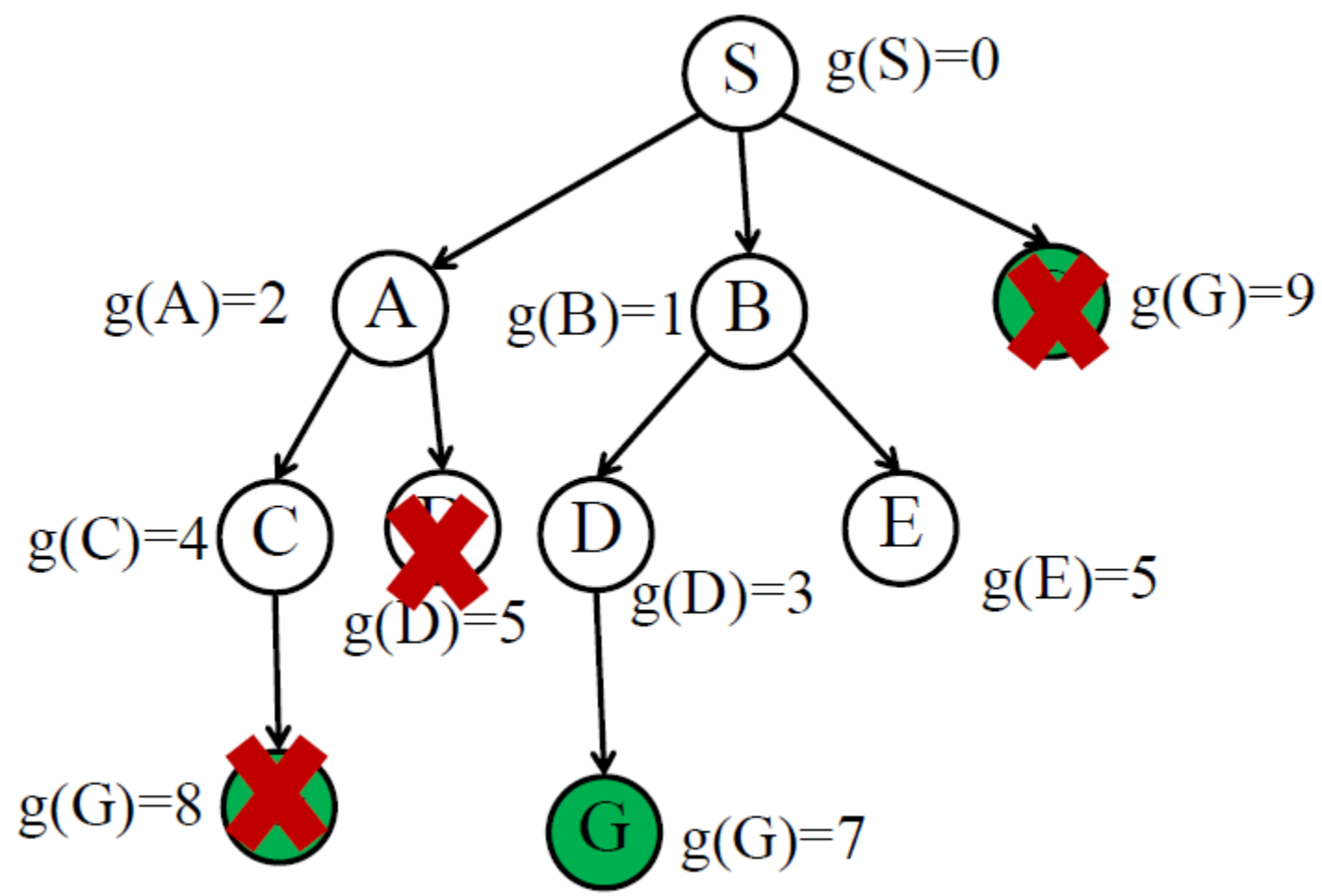


$S \rightarrow A \rightarrow C$





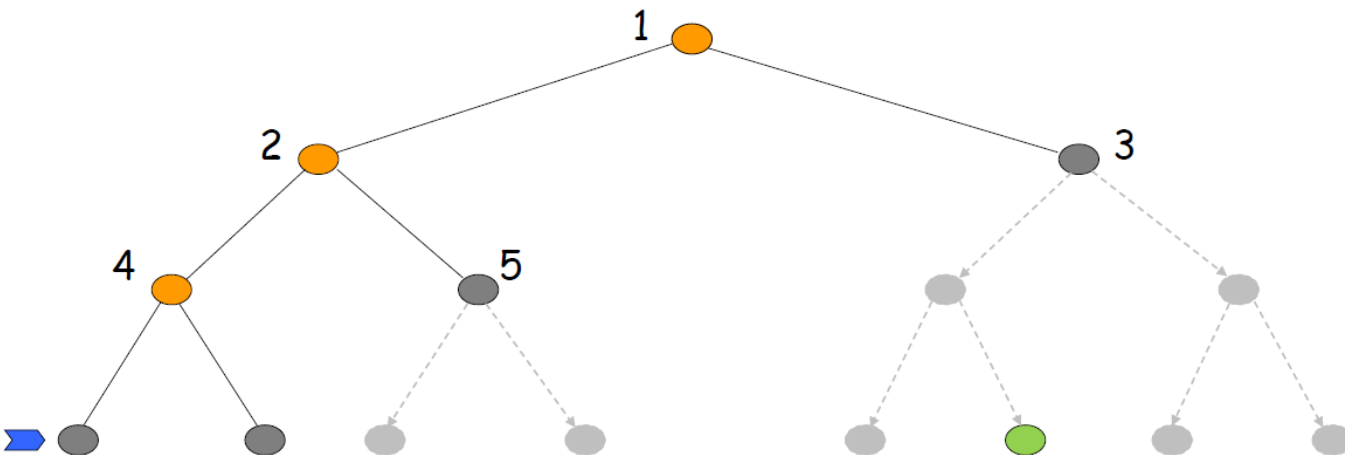
$S \rightarrow B \rightarrow D \rightarrow G$



جستجوی اول عمق - DFS

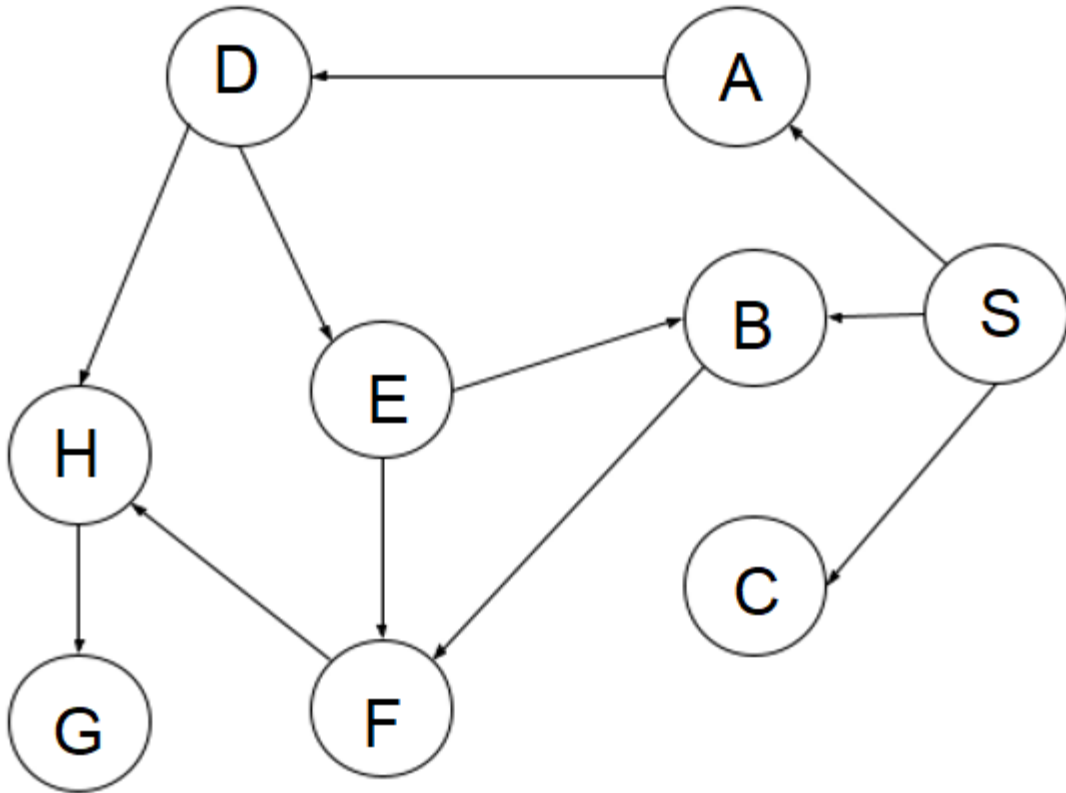
❖ استراتژی : در هر لحظه گره ای برای بسط دادن انتخاب می شود که بیشترین عمق را داشته باشد.

❖ برای پیاده سازی لیست frontier یا fringe از یک پشته استفاده می کند.

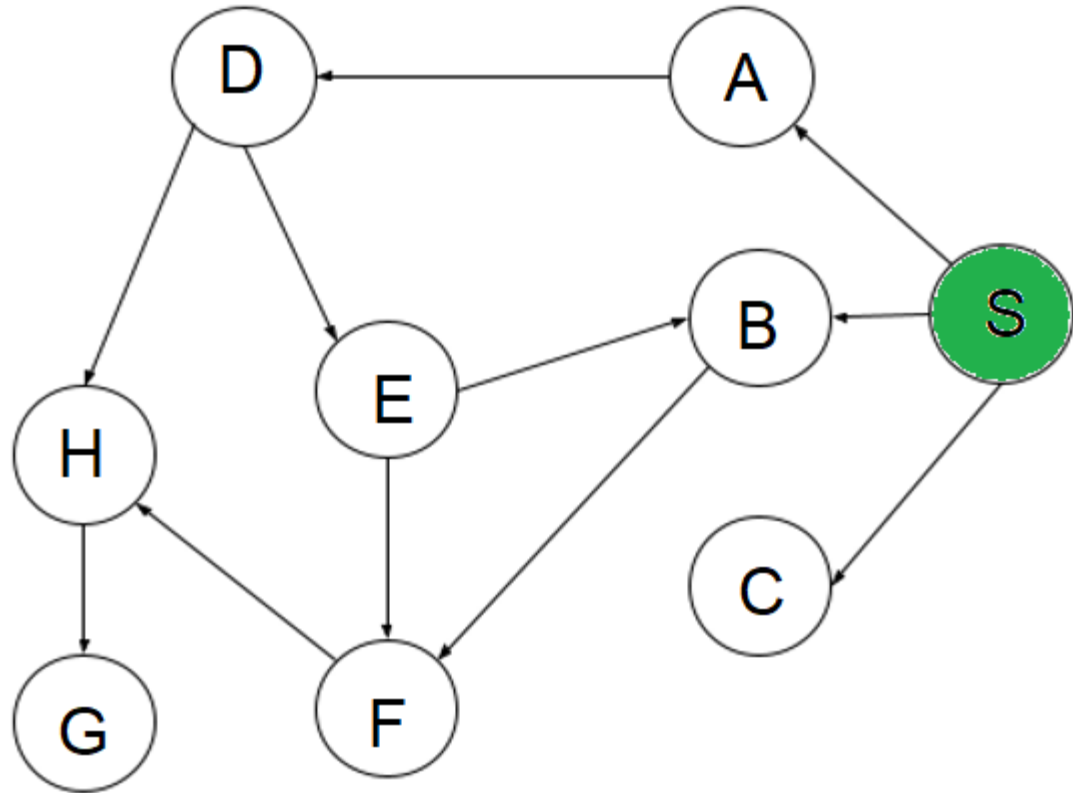


مثال جستجو با استفاده از الگوریتم DFS QA

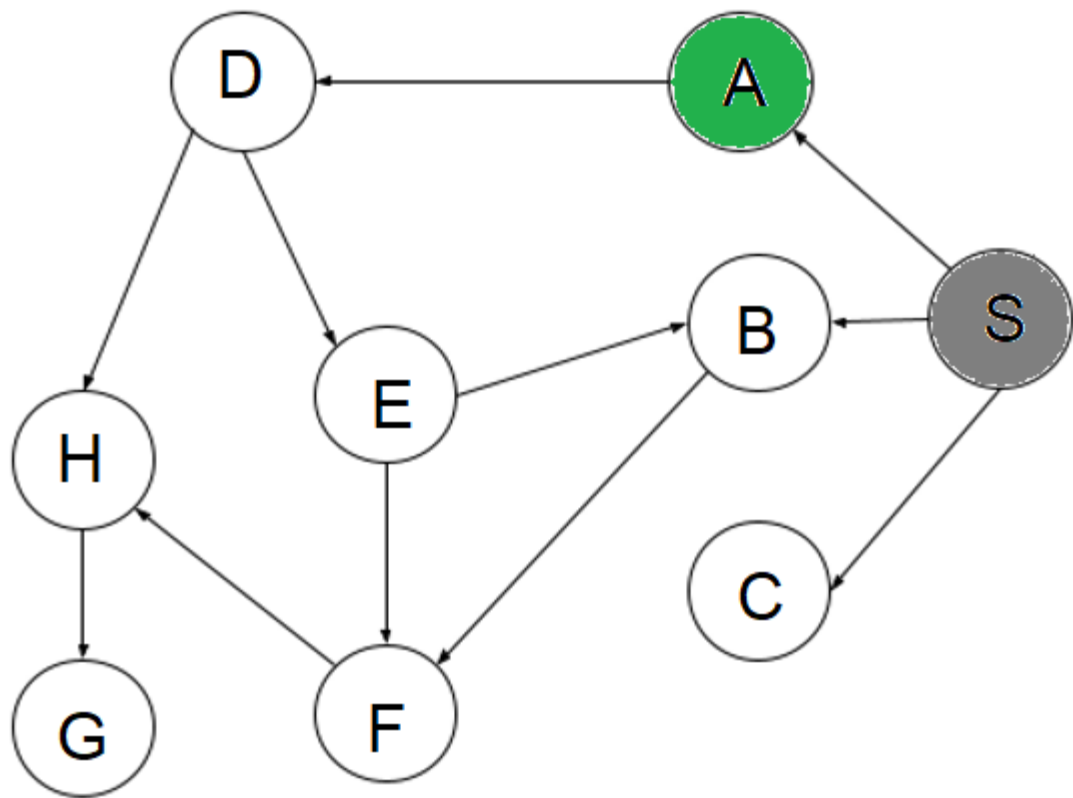
توجه : در موارد یکسان بر اساس ترتیب حروف الفبا عمل کنید.



S



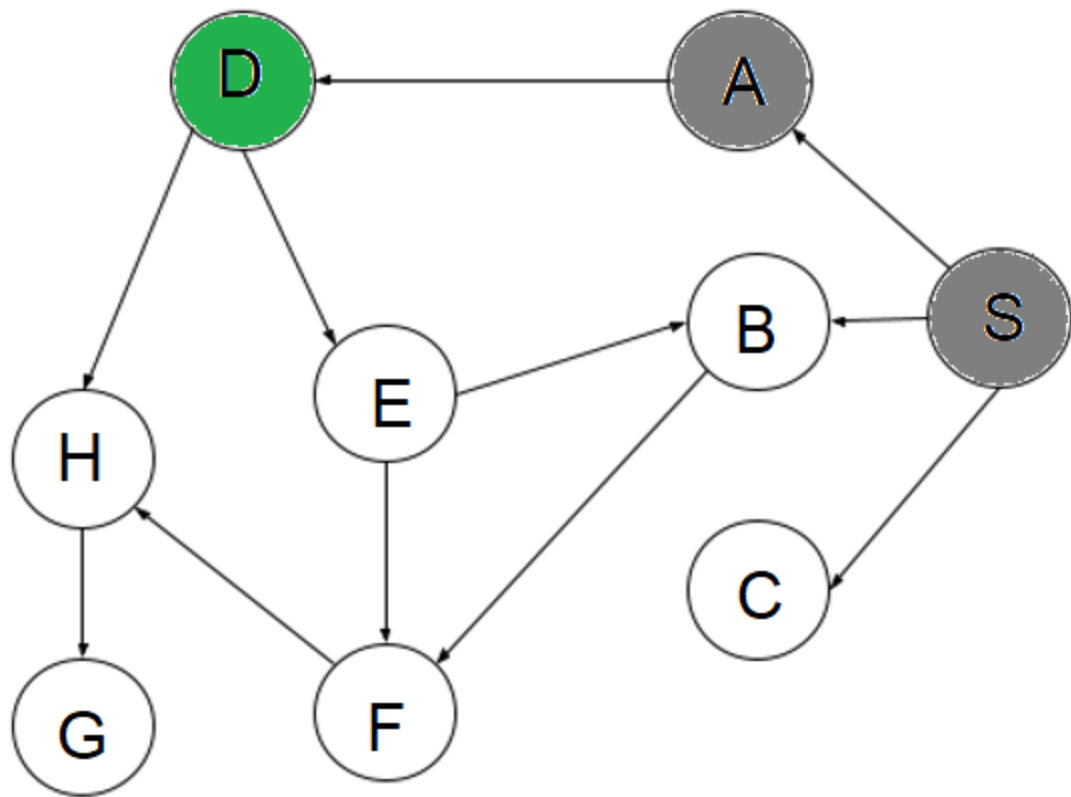
§
 $S \rightarrow A$

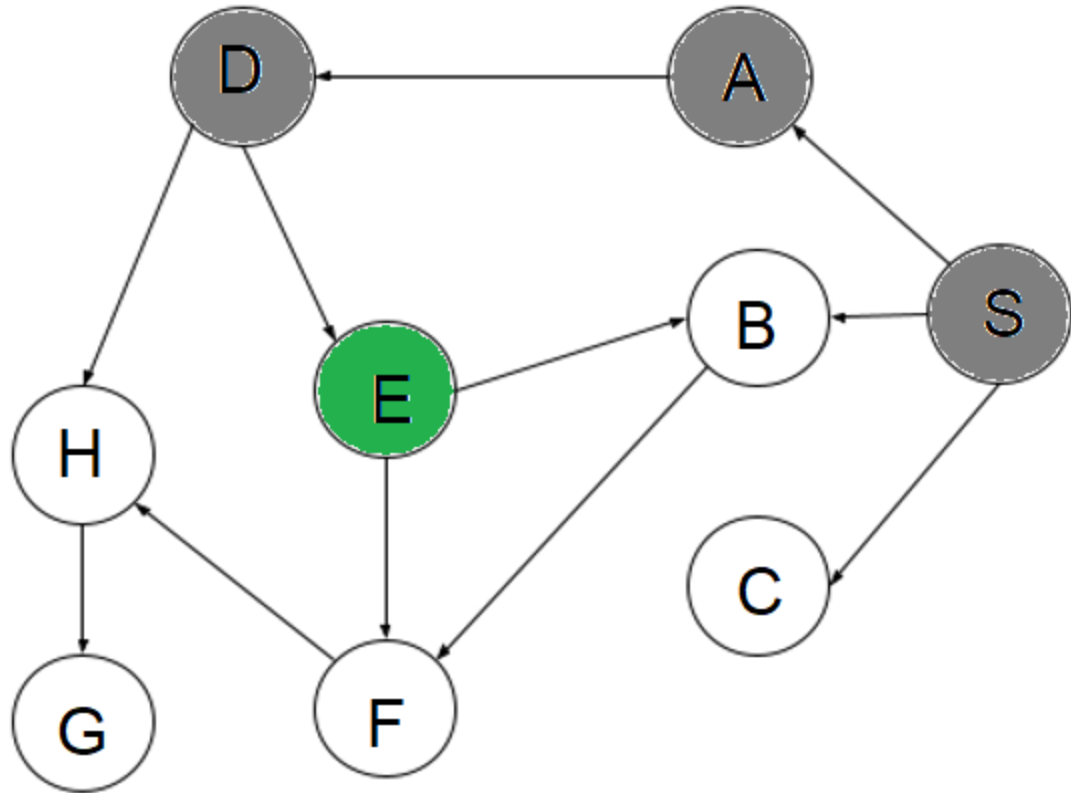


\S

~~$S \rightarrow A$~~

$S \rightarrow A \rightarrow D$



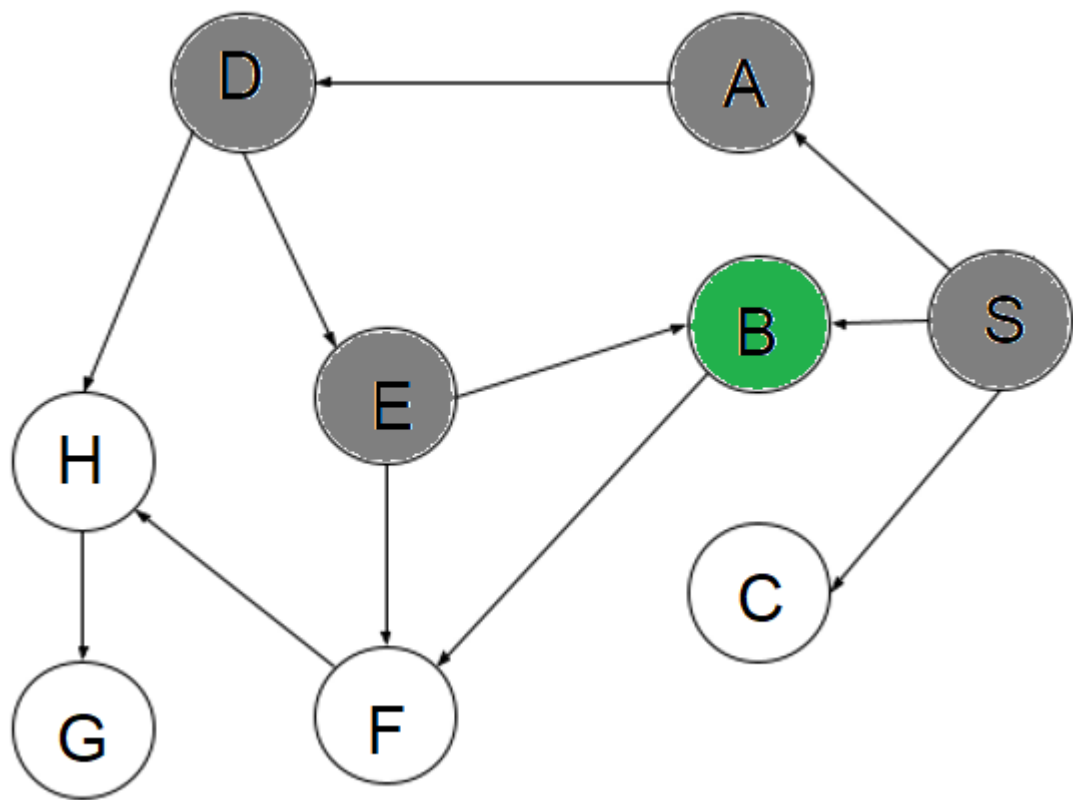


S

~~S~~→A

~~S~~→A→~~D~~

S→A→D→E



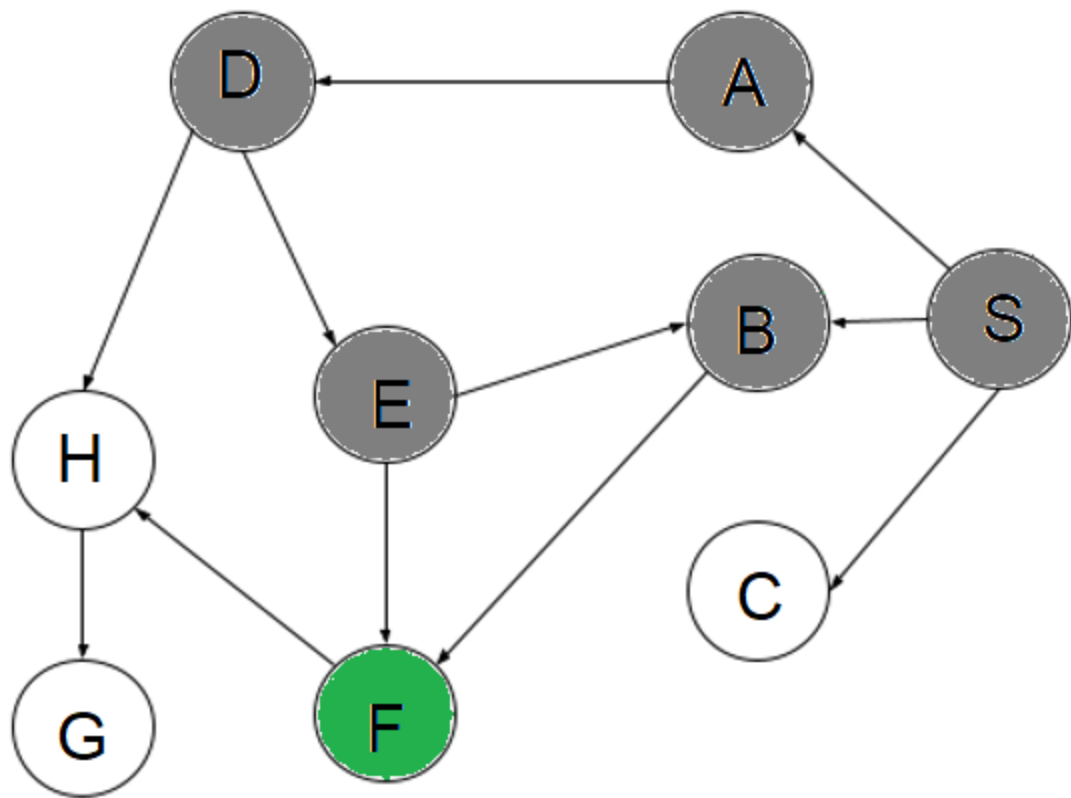
S

~~$S \rightarrow A$~~

~~$S \rightarrow A \rightarrow D$~~

~~$S \rightarrow A \rightarrow D \rightarrow E$~~

$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B$



S

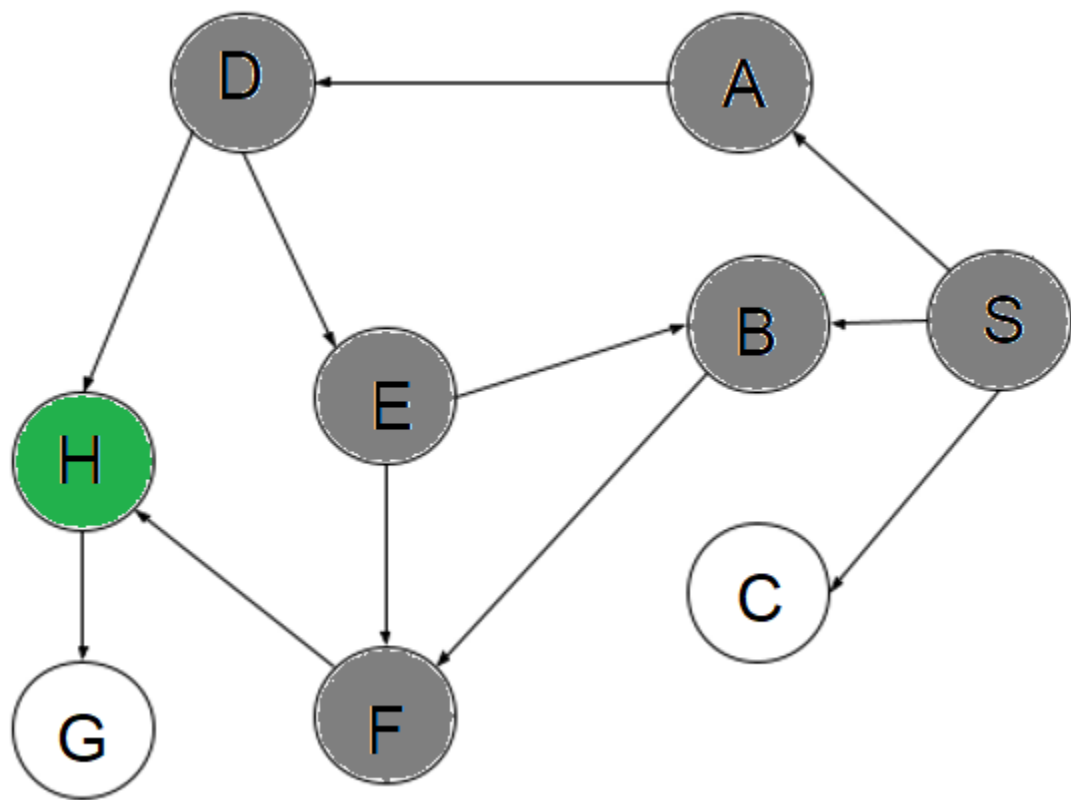
~~$S \rightarrow A$~~

~~$S \rightarrow A \rightarrow D$~~

~~$S \rightarrow A \rightarrow D \rightarrow E$~~

~~$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B$~~

$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B \rightarrow F$



S

~~S → A~~

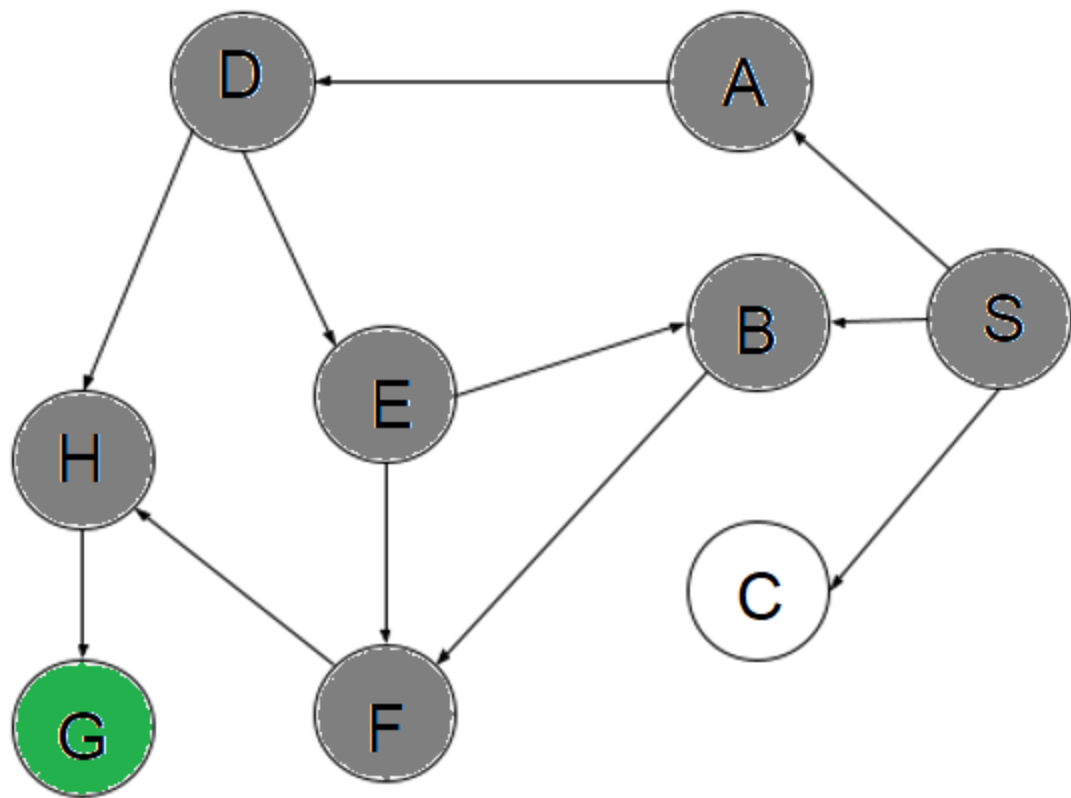
~~S → A → D~~

~~S → A → D → E~~

~~S → A → D → E → B~~

~~S → A → D → E → B → F~~

S → A → D → E → B → F → H



S

$S \rightarrow A$

$S \rightarrow A \rightarrow D$

$S \rightarrow A \rightarrow D \rightarrow E$

$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B$

$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B \rightarrow F$

$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B \rightarrow F \rightarrow H$

$S \rightarrow A \rightarrow D \rightarrow E \rightarrow B \rightarrow F \rightarrow H \rightarrow G$

جستجو های آگاهانه و توابع هیوریستیک



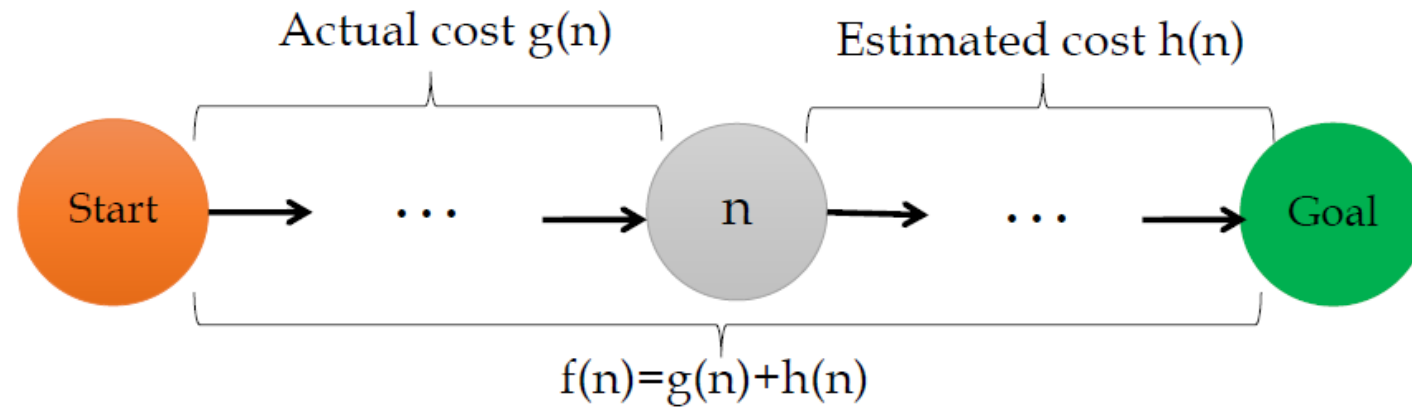
جستجوی A^*

• تابع ارزیاب: $f(n) = g(n) + h(n)$

• $g(n)$ هزینه واقعی مسیر از شروع تا گره n

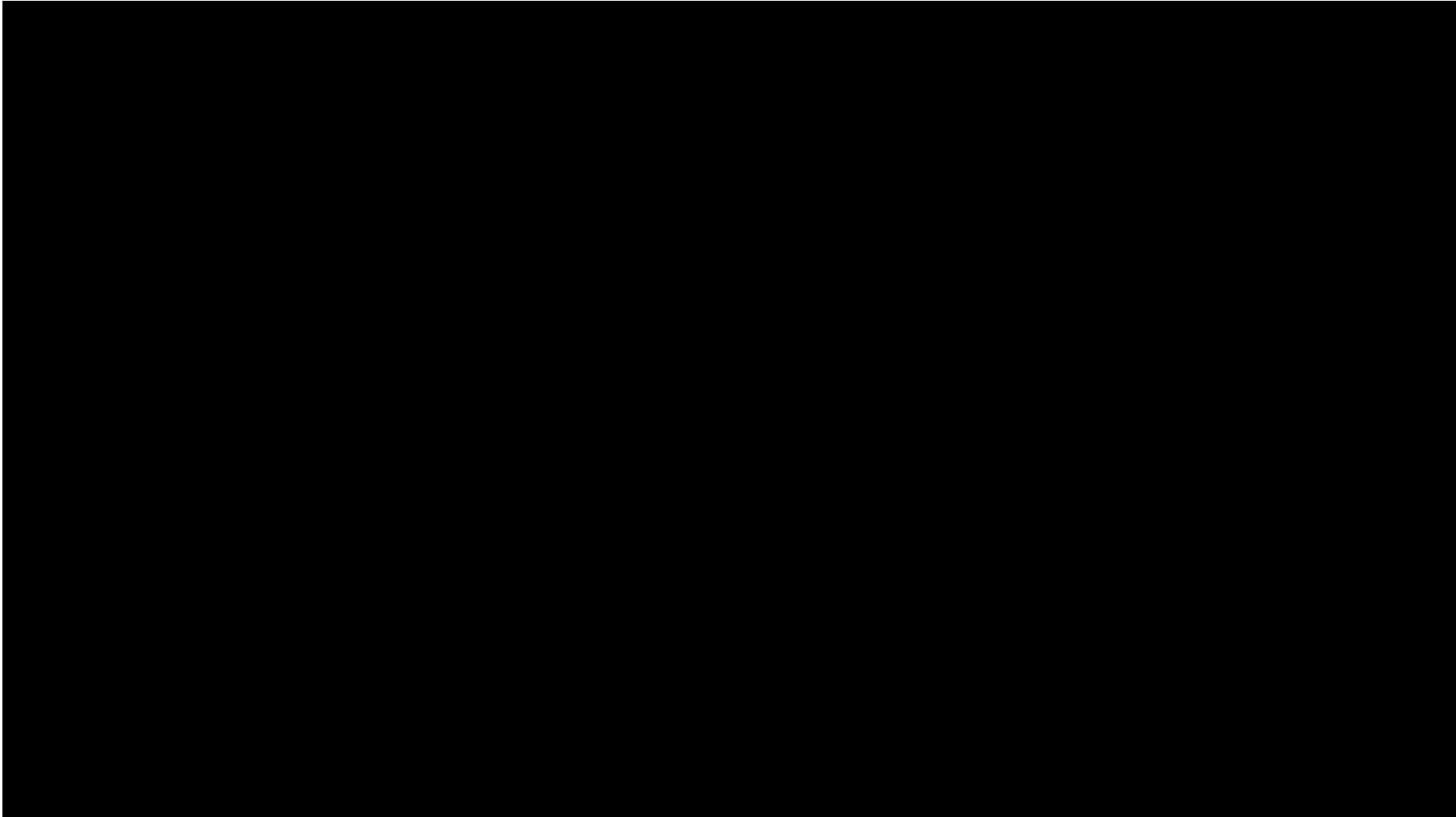
• $h(n)$ هزینه تخمینی ارزان‌ترین مسیر از نود n تا هدف

• $f(n)$ هزینه تخمینی کل مسیر از ریشه درخت تا گره هدف از طریق گره n



ویدیوی آموزش الگوریتم A^* همراه با مثال

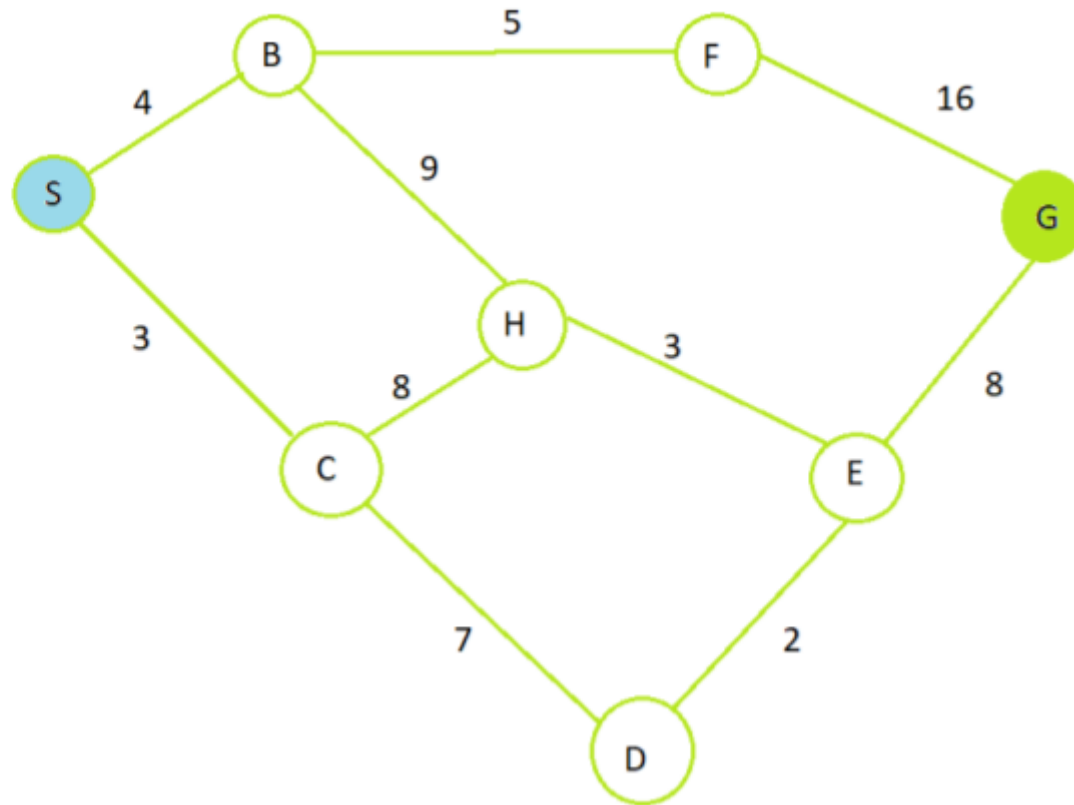
ویدیوی زیر در این [لینک](#) قابل مشاهده است.



مثال جستجو با استفاده از الگوریتم A^* QA

توجه : در موارد یکسان بر اساس ترتیب حروف الفبا عمل کنید.

Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



الف) به ازای چه مقادیری از $h(E)$ هیوریستیک داده شده هم قابل قبول (admissible) و هم سازگار (consistent) است؟

❖ شرط قابل قبول بودن : $h(n) \leq h^*(n)$

پس برای قابل قبول بودن می بایست $h(E) \leq 8$

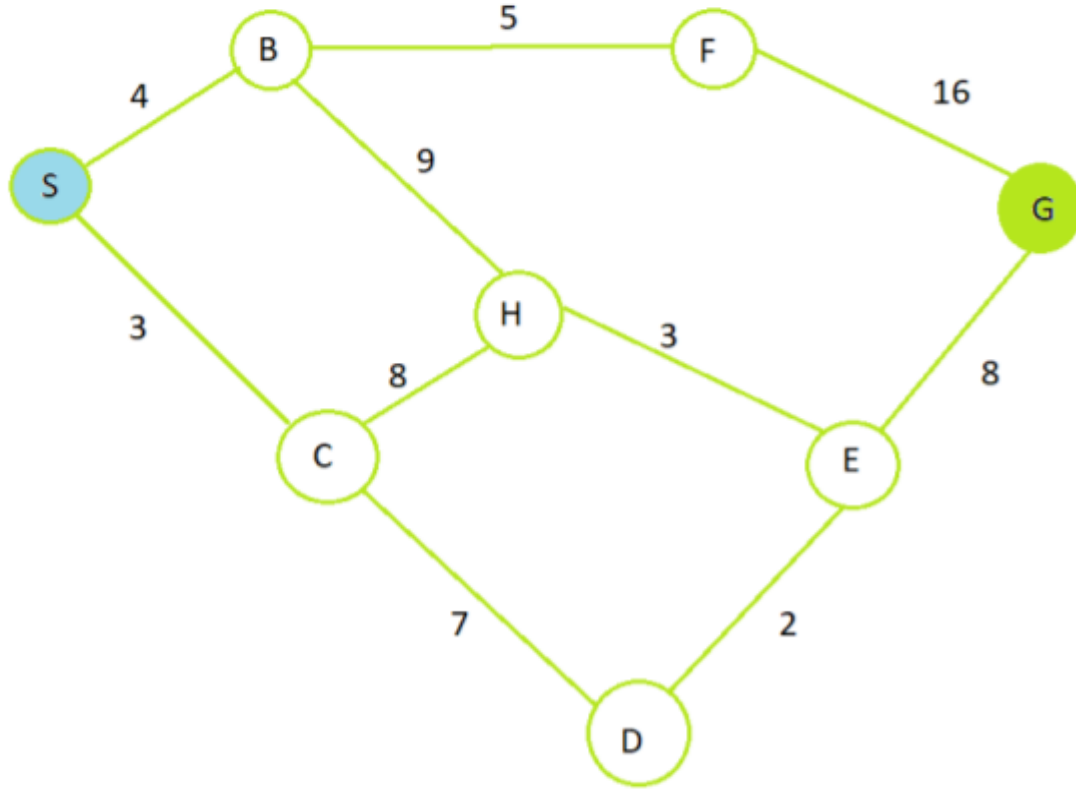
❖ شرط سازگار بودن : $h(n) \leq c(n,a,n') + h(n')$

$$h(D) \leq c(D, E) + h(E) = 2 + h(E) \rightarrow h(E) \geq 4$$

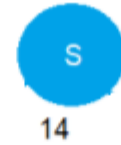
$$h(H) \leq c(H, E) + h(E) = 3 + h(E) \rightarrow h(E) \geq 3$$

پس میبایست $4 \leq h(E) \leq 8$

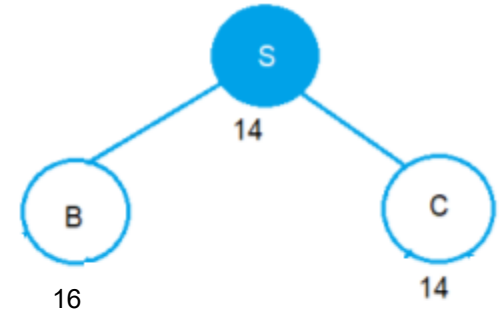
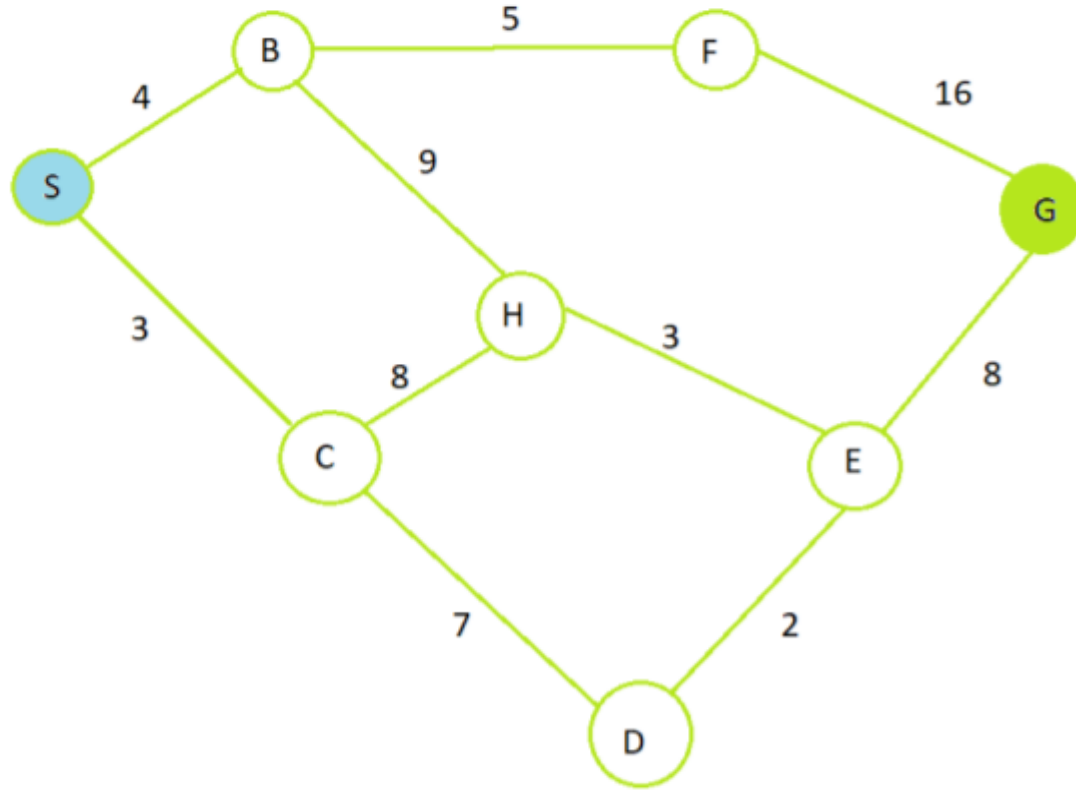
Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



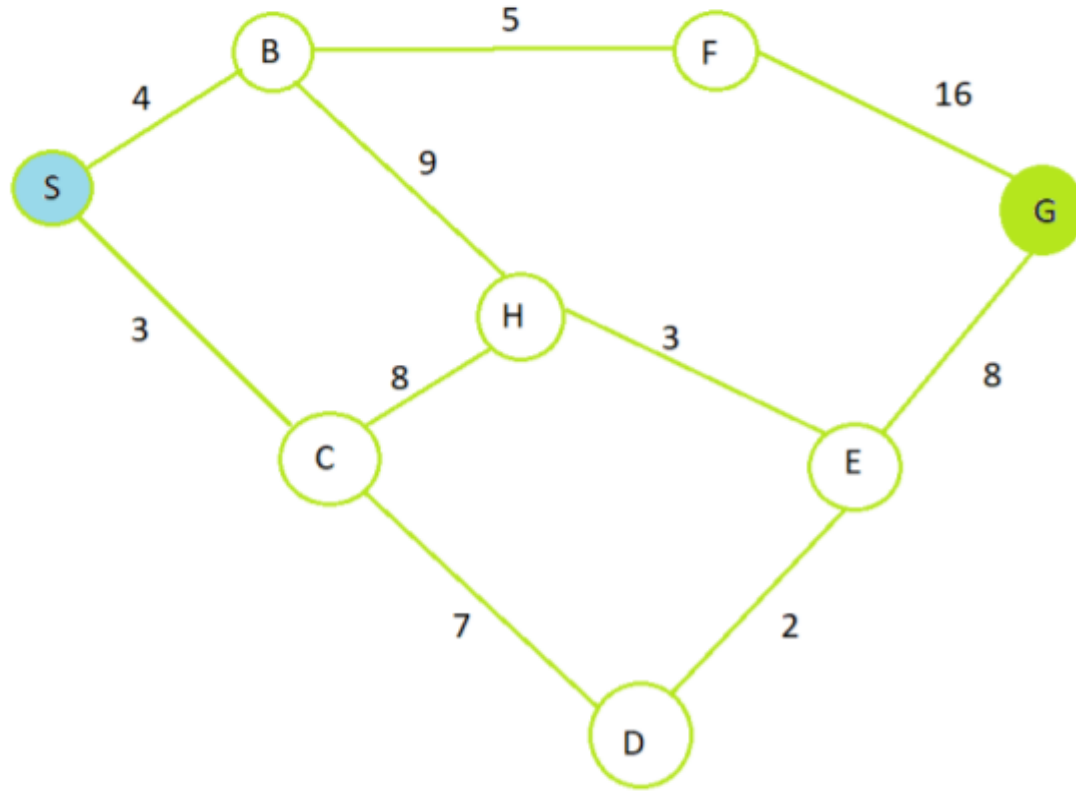
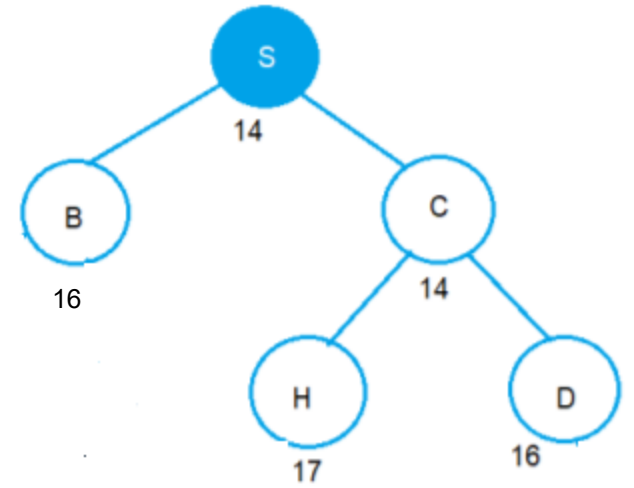
(ب) به ازای $h(E) = 6$ جستجوی
گرافی A^* را برای گراف فضای حالت
داده شده اجرا کنید.

[illegible]

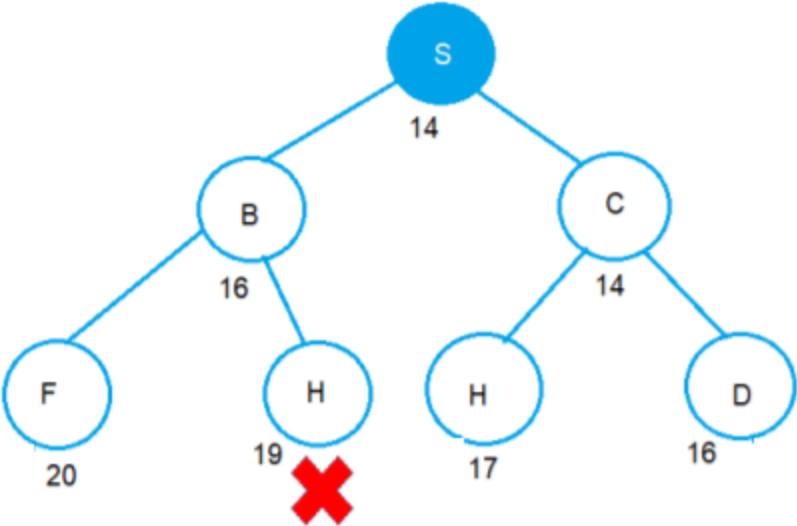
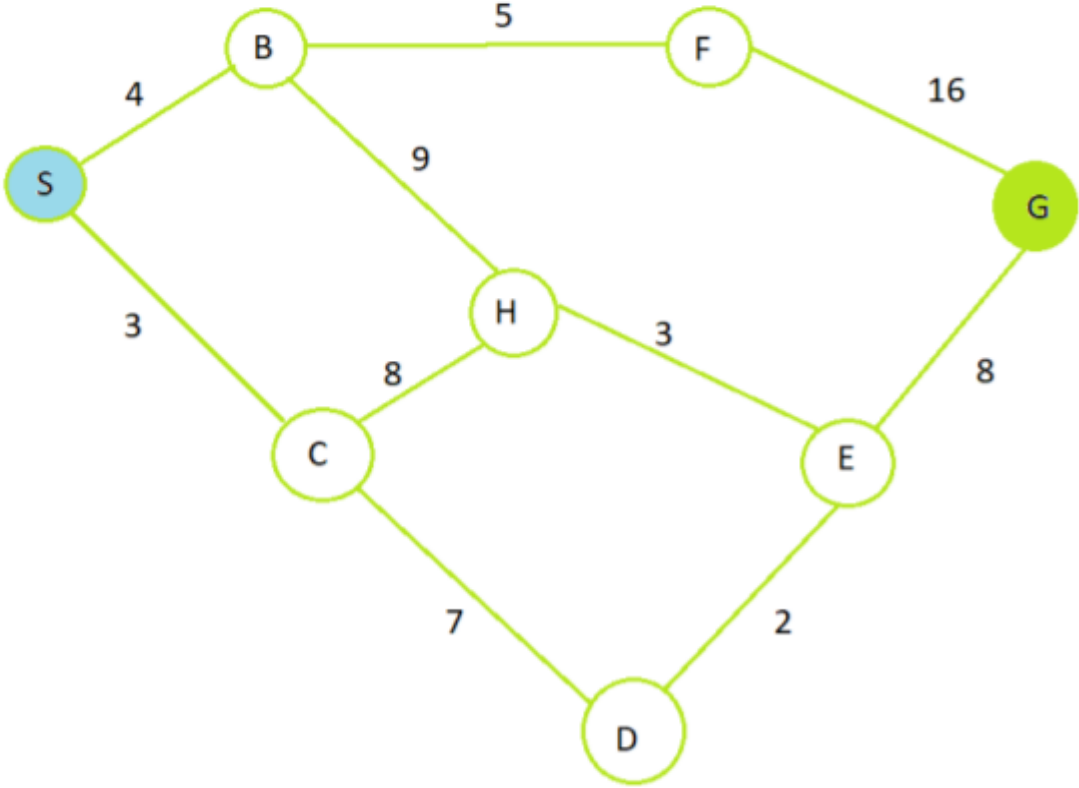
Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6

[illegible]

Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6

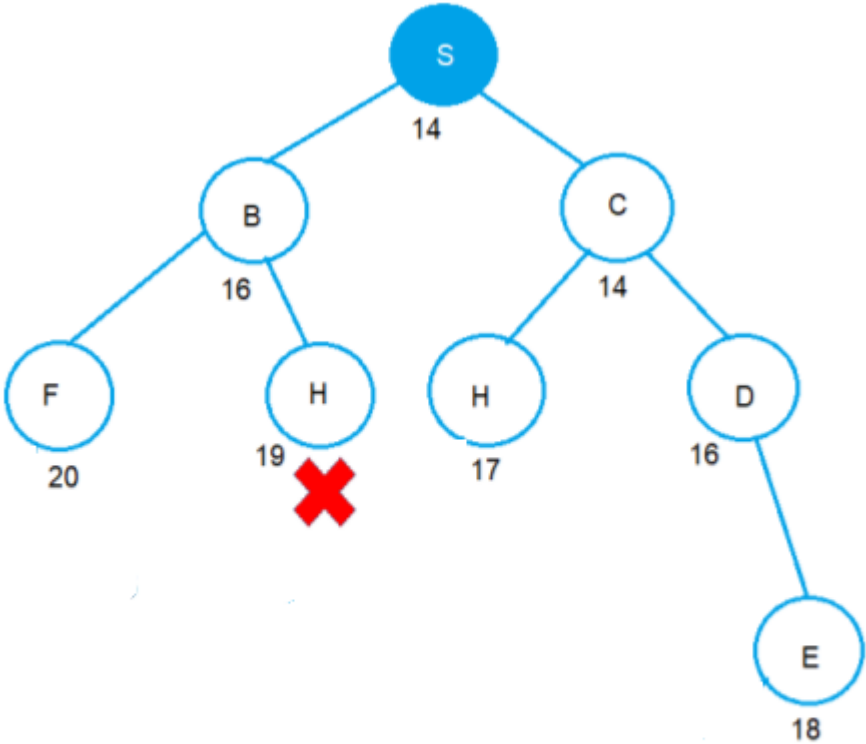
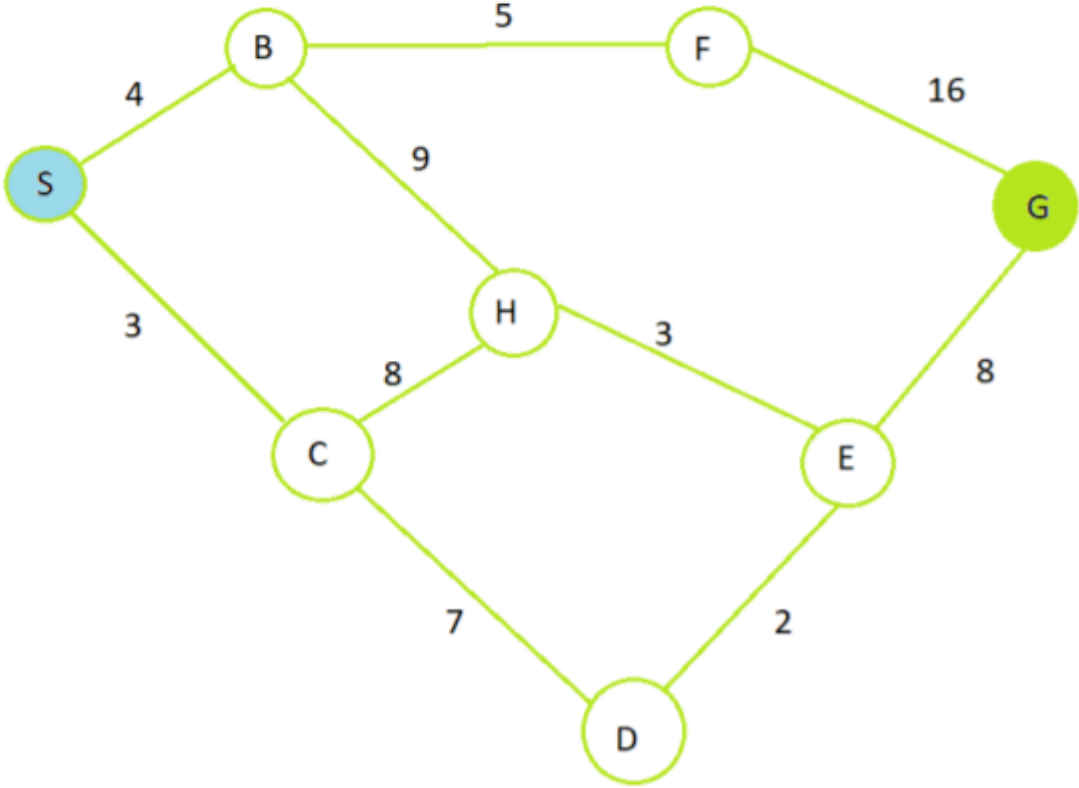
[illegible]

Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



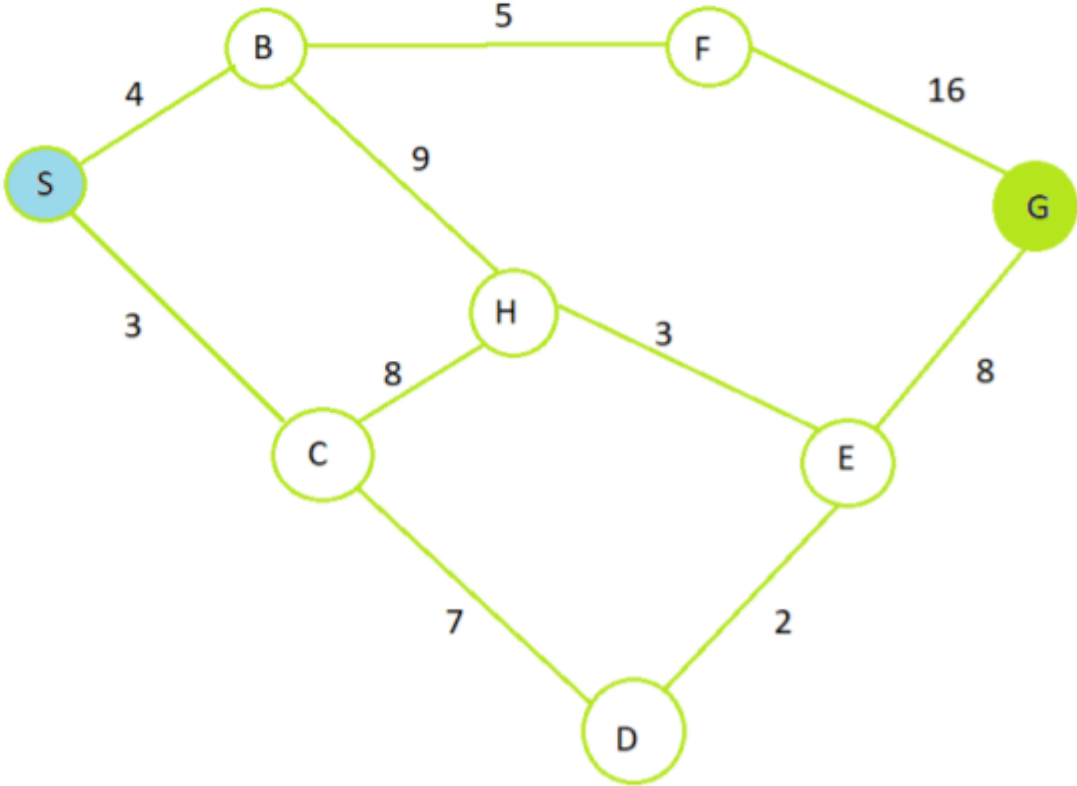
Frontier	Explored
S	-
B, C	S
B, H, D	S, C
H, D, F	S, C, B

Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6

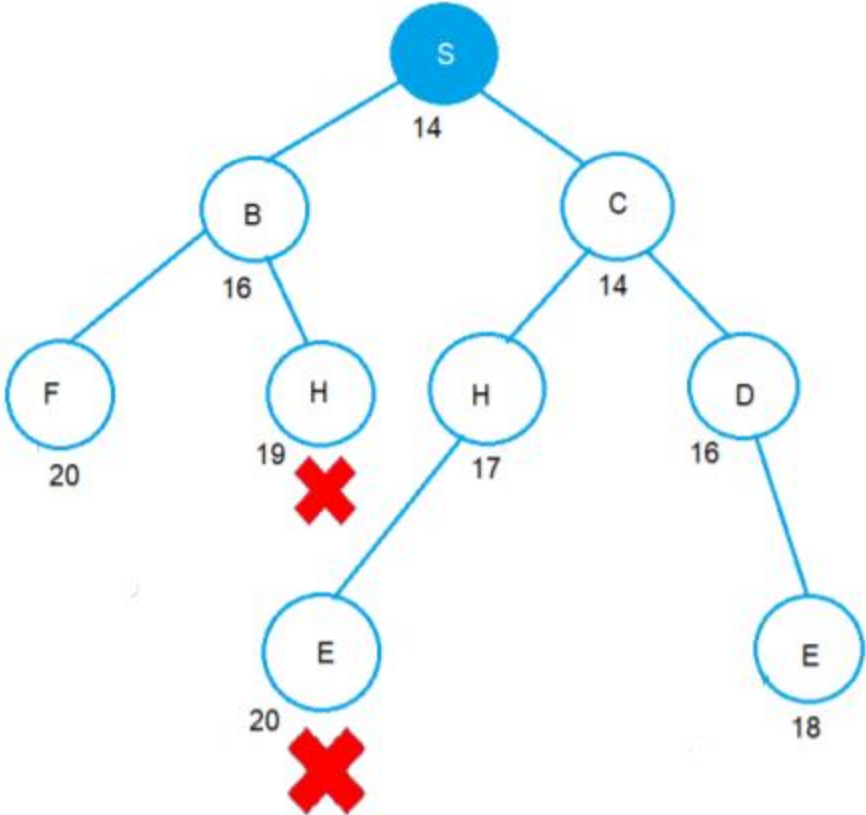


Frontier	Explored
S	-
B, C	S
B, H, D	S, C
H, D, F	S, C, B
H, F, E	S, C, B, D

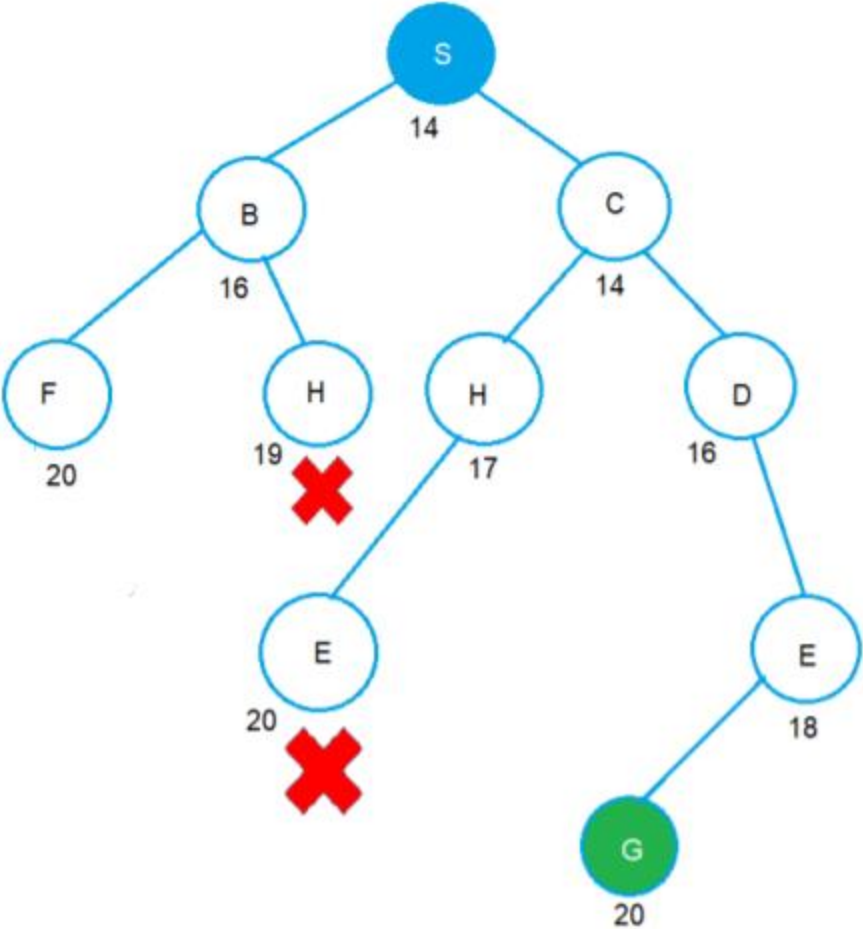
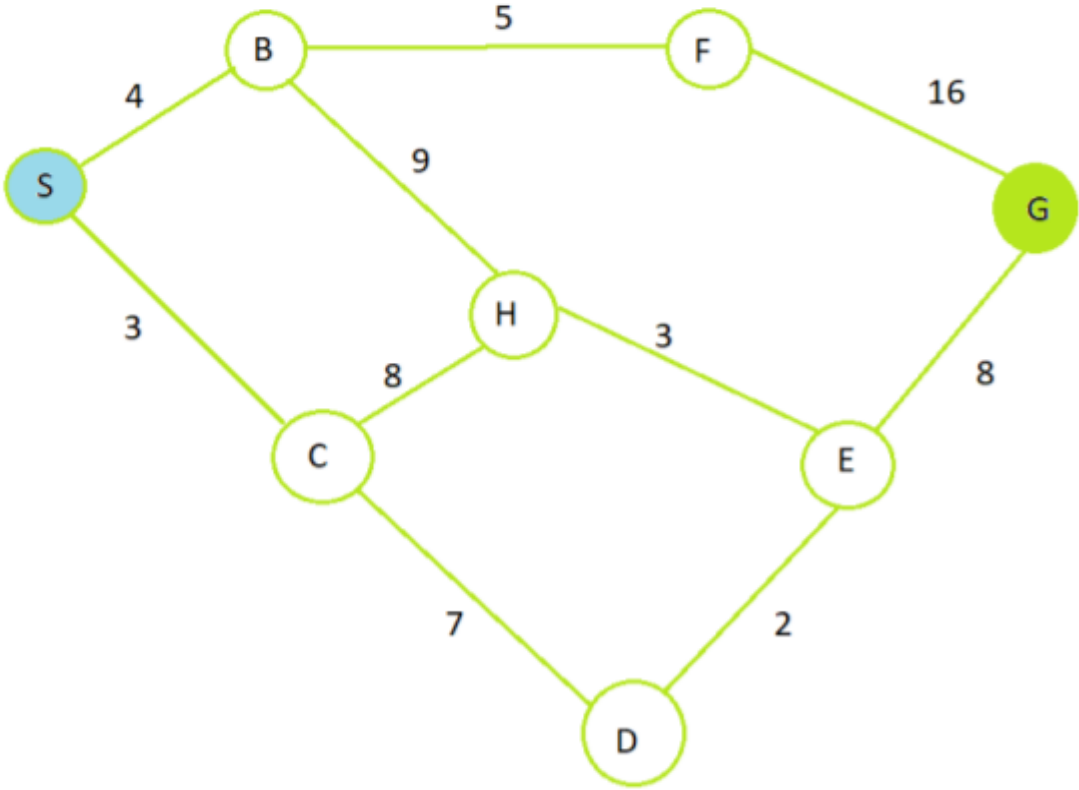
Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



Frontier	Explored
S	-
B, C	S
B, H, D	S, C
H, D, F	S, C, B
H, F, E	S, C, B, D
F, E	S, C, B, D, H

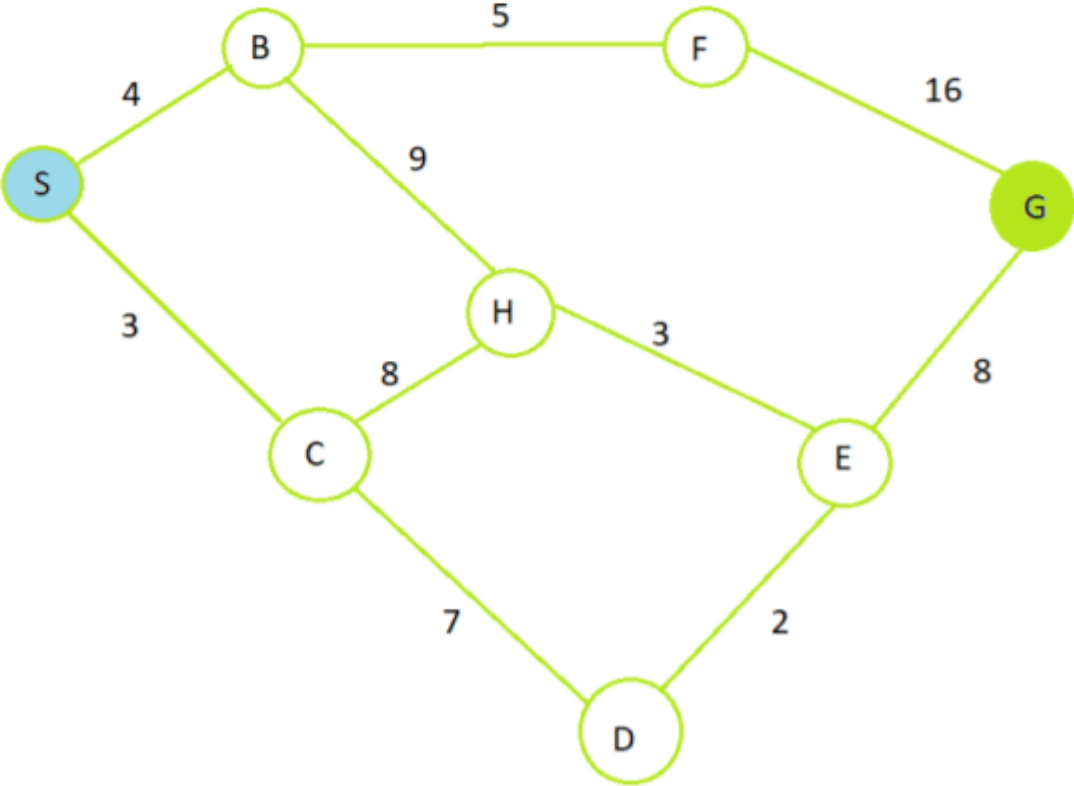


Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6

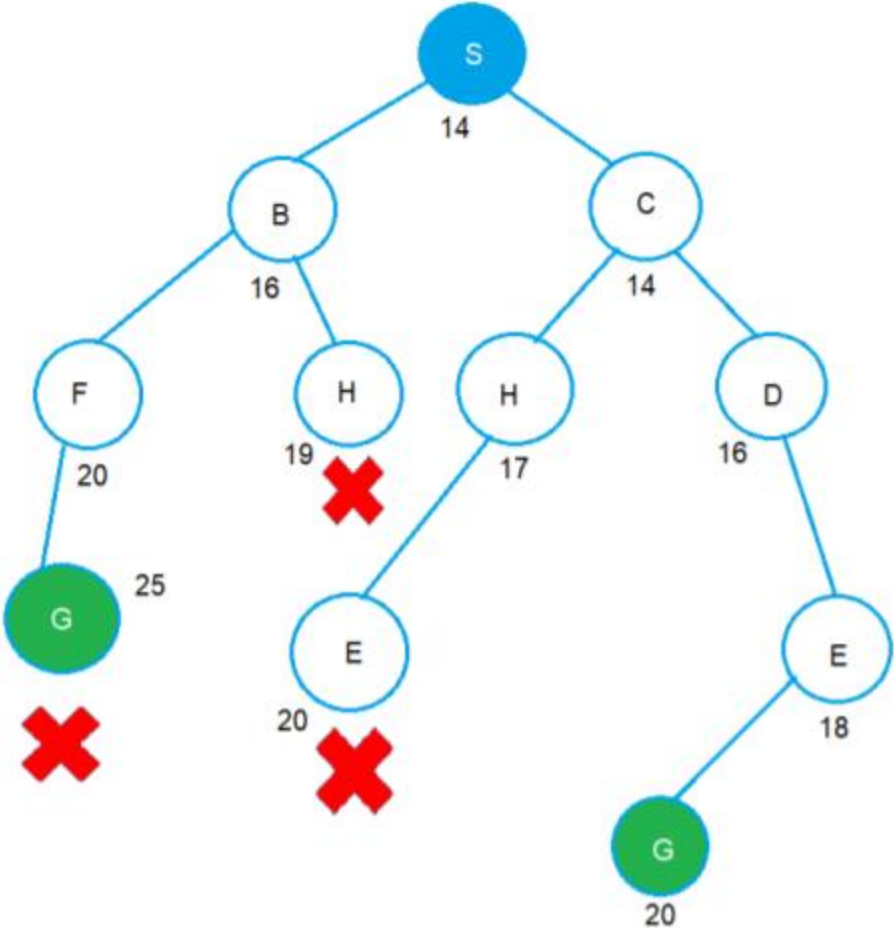


Frontier	Explored
S	-
B, C	S
B, H, D	S, C
H, D, F	S, C, B
H, F, E	S, C, B, D
F, E	S, C, B, D, H
F, G	S, C, B, D, H, E

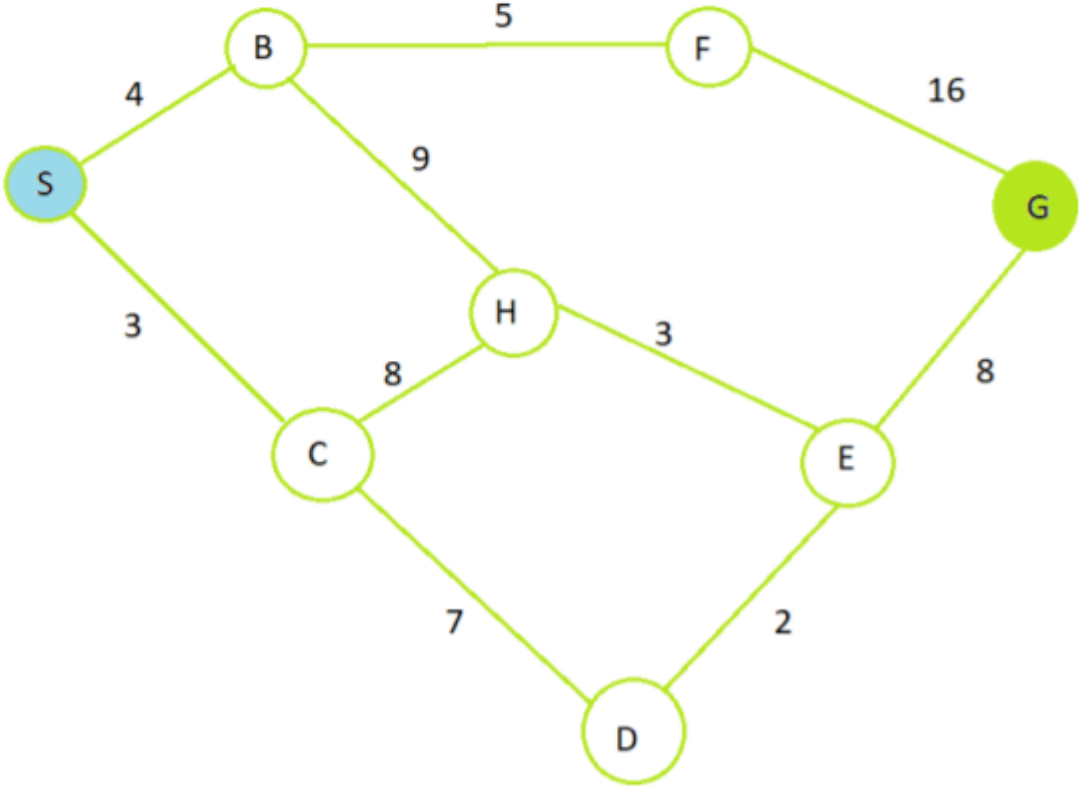
Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



Frontier	Explored
S	-
B, C	S
B, H, D	S, C
H, D, F	S, C, B
H, F, E	S, C, B, D
F, E	S, C, B, D, H
F, G	S, C, B, D, H, E
G	S, C, B, D, H, E, F



Node	h
A	14
B	12
C	11
D	6
E	?
F	11
G	0
H	6



Frontier	Explored
S	-
B, C	S
B, H, D	S, C
H, D, F	S, C, B
H, F, E	S, C, B, D
F, E	S, C, B, D, H
F, G	S, C, B, D, H, E
G	S, C, B, D, H, E, F
	S, C, B, D, H, E, F, G

