

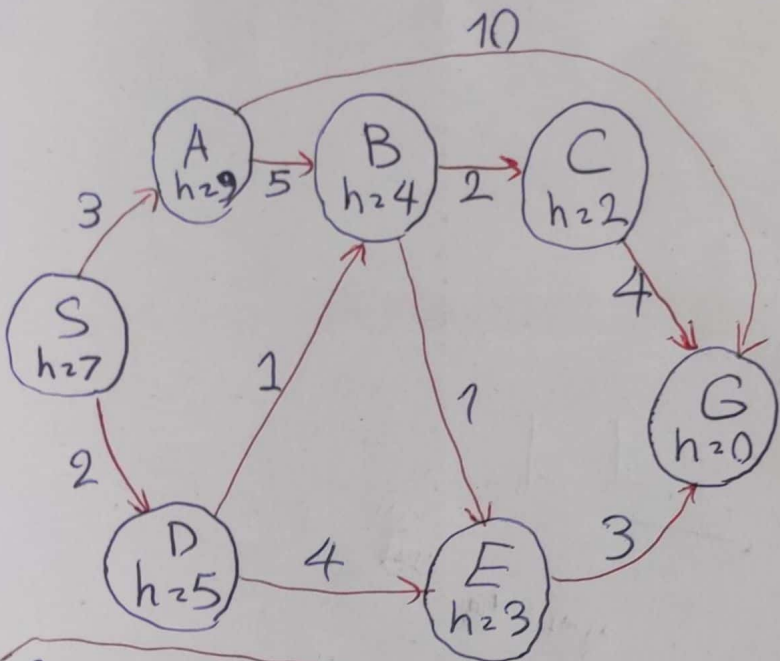
۹۹۵۲۱۲۷۱

HW3_AI

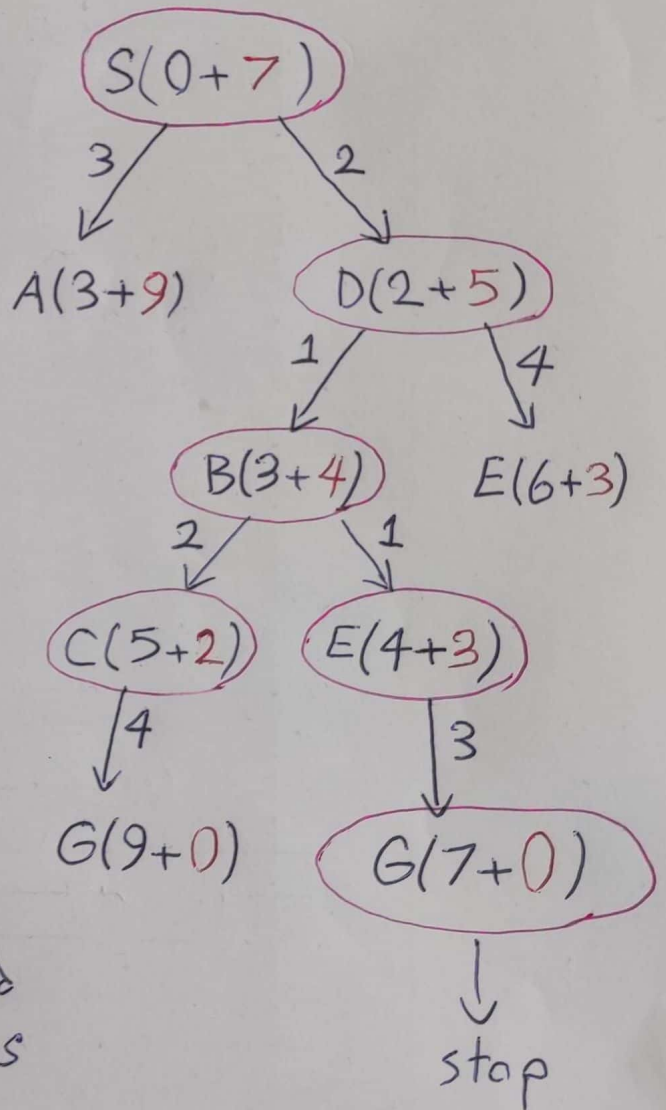
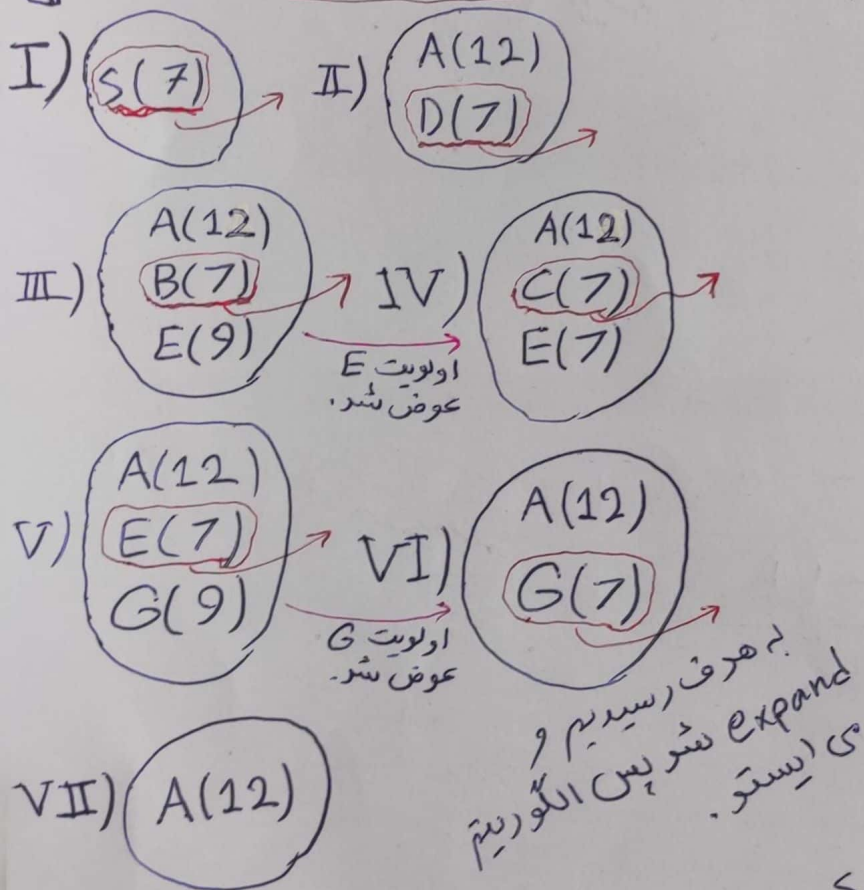
فرزان رحمانی

سؤال یک (الف)

چون باتوجه به اعداد داده شده تابع h $admissible$ و $consistent$ است
لذا از A^* Graph Search استفاده می کنیم.



frontier: priority queue
 $f(n) = h(n) + g(n)$ بر اساس



مسیر بهینه: $S \xrightarrow{2} D \xrightarrow{1} B \xrightarrow{1} E \xrightarrow{3} G$

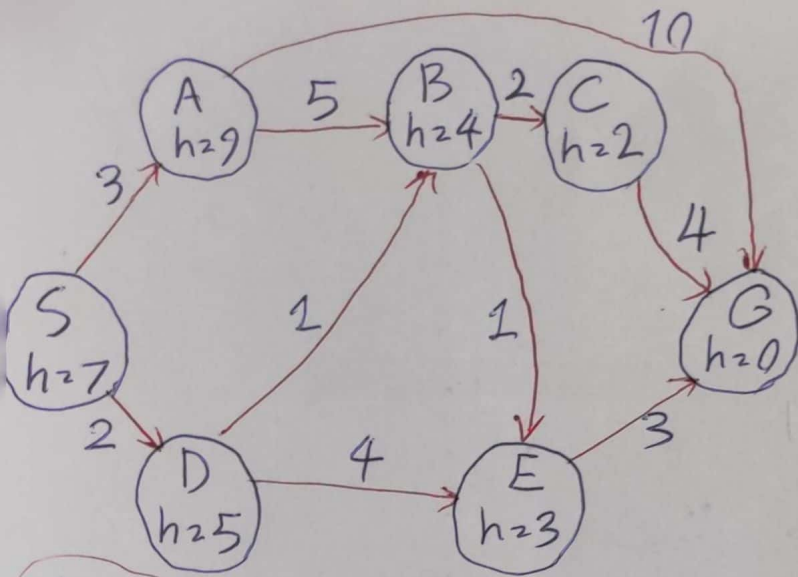
طول مسیر بهینه: 7 ← کوتاه ترین مسیر را یافت.

ترتیب گره های expand شده: S, D, B, C, E, G

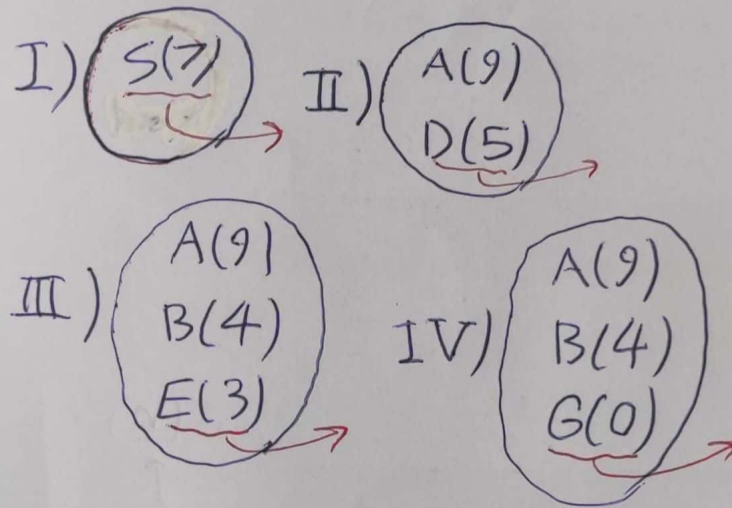
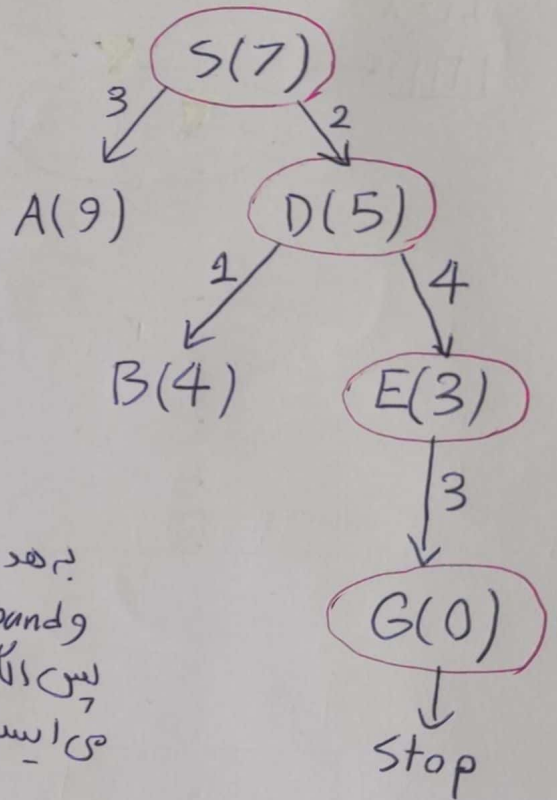
Optimality A^* { Tree Search: heuristic should be admissible
Graph Search: heuristic should be consistent

سؤال دو (ب) Greedy Search

Strategy: expand a node that you think is closest to a goal state according to the heuristic function.



frontier: priority queue
بر اساس $h(n)$



به هدف رسیدیم
و expand شد
پس الگوریتم
می ایستد.

مسیر یافته شده: $S \xrightarrow{2} D \xrightarrow{4} E \xrightarrow{3} G$

طول مسیر: 9 ← جواب optimal نیست زیرا مسیر با طول 7 هم وجود دارد.
ترتیب گره های expand شده: S, D, E, G (گره های کمتری را expand کرد و سریع تر از A* عمل کرد.)

مزایا و معایب: الگوریتم Greedy Search می تواند بسیار سریع عمل کند همان طور که در مثال بالا دیدیم از A* سریع تر بود و معمولاً هم بسیار سریع است ولی در worst-case مثل یک badly-guided DFS است. همچنین جواب optimal را هم پیدا نمی کند مانند مثال بالا که کوتاه ترین مسیر را پیدا نکرد. ولی الگوریتم A* معمولاً سرعت کمتری نسبت به Greedy دارد و node های بیشتری را می بیند و expand می کند ولی همیشه جواب optimal و بهینه را پیدا می کند مانند مثال بالا که کوتاه ترین مسیر را پیدا کرد. به شرط انتخاب تابع از نظر حافظه هر دو از priority queue استفاده می کنند که شبیه هم است ولی معمولاً چون greedy گره های کمتری را می بیند پیچیدگی کمتری دارد و حافظه کمتری نیاز دارد. خوب برای h

همچنین Greedy Search الگوریتم کاملی نیست و ممکن است مسیری به هدف پیدا نکند ولی A^* Complete است.

not complete

Greedy Search چون node هایی که قبلاً expand شده اند را نیاز ندارد ذخیره کند می تواند

پیچیدگی حافظه ای چقدر باشد و حافظه کمتری نسبت به A^* می گیرد.

A^* نیاز دارد تمامی node های دیده شده را در حافظه نگه دارد.

proportional to the number of nodes in the fringe and the length of the found path

A^* زمانی که فضای جستجو خیلی بزرگ باشد بی فایده است چون حافظه زیادی نیاز دارد.

Search	Frontier	Completeness	Optimally	Time	Space	d = depth of the solution
A^*	Priority Queue	YES	Yes	$O(b^d)$	$O(b^d)$	
Greedy Search	Priority Queue	NO	NO	$O(b^m)$	$O(b^m)$	

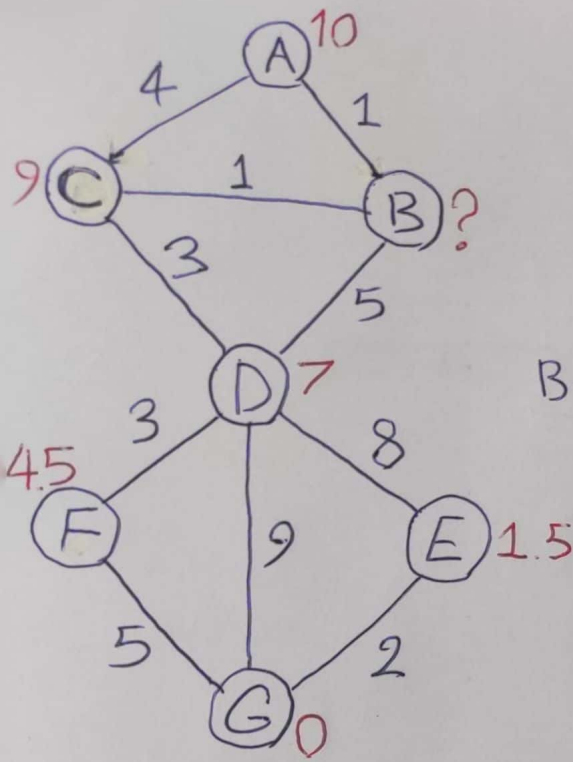
m = maximum depth of the search space

سؤال دو

الف) a heuristic h is admissible if:

$$0 \leq h(n) \leq h^*(n)$$

true cost to a nearest goal



کوتاه ترین مسیر از B به G: $B \xrightarrow{1} C \xrightarrow{3} D \xrightarrow{3} F \xrightarrow{5} G$

چون طول این مسیر $1+3+3+5=12$ است پس

$h^*(B)=12$ است. لذا $h_1(B)$ می تواند یک عدد از 0 تا 12 باشد.

$$0 \leq h_1(B) \leq 12$$

ب)

Consistency: heuristic arc cost \leq actual cost for each arc

حال این تعریف را برای

بال های متصل به B می نویسیم

e.g. $h(A) \xrightarrow{c(A,C)} h(C) : h(A) - h(C) \leq c(A,C)$

$$\begin{aligned} h_1(A) - h_1(B) &\leq 1 \rightarrow 10 - h_1(B) \leq 1 \rightarrow h_1(B) \geq 9 \\ h_1(B) - h_1(A) &\leq 1 \rightarrow h_1(B) - 10 \leq 1 \rightarrow h_1(B) \leq 11 \\ h_1(B) - h_1(C) &\leq 1 \rightarrow h_1(B) - 9 \leq 1 \rightarrow h_1(B) \leq 10 \\ h_1(C) - h_1(B) &\leq 1 \rightarrow 9 - h_1(B) \leq 1 \rightarrow h_1(B) \geq 8 \\ h_1(B) - h_1(D) &\leq 5 \rightarrow h_1(B) - 7 \leq 5 \rightarrow h_1(B) \leq 12 \\ h_1(D) - h_1(B) &\leq 5 \rightarrow 7 - h_1(B) \leq 5 \rightarrow h_1(B) \geq 2 \end{aligned}$$

همه این شرط ها باید

همزمان برقرار

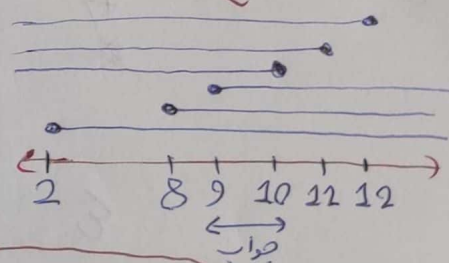
باشند پس بین

آنها اشتراک می گیریم.

استخراج

$$9 \leq h(B) \leq 10$$

لذا $h_1(B)$ باید یک عدد از 9 تا 10 باشد.



$A \rightarrow C \rightarrow B \rightarrow D$

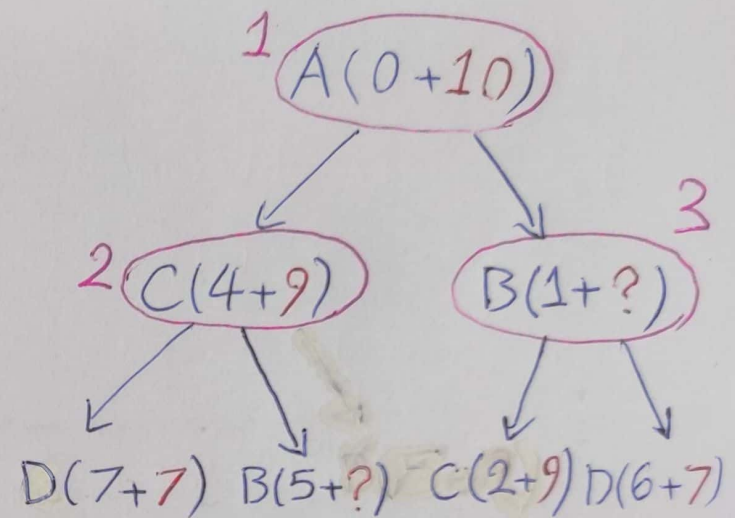
ج) برای اینکه C قبل از B دیده شود باید $f(C) < f(B)$ پس $h(C) + g(C) < h(B) + g(B)$ باشد.

$$\underbrace{9 + 4}_{g(A)+4} < \underbrace{h_1(B) + 1}_{g(A)+1} \Rightarrow h_1(B) > 12$$

برای اینکه B قبل از D دیده شود باید $f(B) < f(D)$
 پس $h(B) + g(B) < h(D) + g(D)$ باشد.

$$h_1(B) + 1 < 7 + 7$$

$$h_1(B) < 13$$



پس $12 \leq h_1(B) \leq 13$ باشد ولی چون طبق قسمت الف اگر بخواهیم کم
 $h_1(B) = 12$ باشد لذا $12 \leq h_1(B) \leq 12$ باشد پس $h_1(B) = 12$ می شود.
 اگر هم بخواهیم h_1 Consistent باشد مقدار قابل قبولی برای این ترتیب پیدا نمی شود.

حال از دو مورد بالا اشتراک می گیریم:

$$\left. \begin{matrix} h_1(B) > 12 \\ h_1(B) < 13 \end{matrix} \right\} \Rightarrow 12 < h_1(B) < 13$$