

جستجوی آگاهانه

پرسش: جستجوی آگاهانه چیست؟ آگاهی در این جستجو به چه چیزی اشاره دارد؟ چه مزیتی نسبت به جستجوی ناآگاهانه دارد؟

روش های مختلف جستجوی آگاهانه

- 1- Greedy Search, 2- A* Search, 3- IDA* Search, 4- MA* Search, 5- SMA* Search, 6- RBFS 7- Hill-climbing, 8- Simulated Annealing, 9- Genetic Algorithm 10-....

نکته: جستجوی آگاهانه، جستجوی هیوریستیک یا جستجوی یافتاری هم نامیده می شود.

پرسش: هیوریستیک چیست؟

مثال: هیوریستیک نزدیکترین همسایه برای مساله فروشنده دوره گرد

1. Select a starting city.
2. Select the one closest to the current city.
3. Repeat step 2 until all cities have been visited.

$O(n^2)$ vs. $O(n!)$

جستجوی هیوریستیک

روشهای جستجوی آگاهانه یا هیوریستیک، از تابع ارزیابی جهت برآورد میزان مطلوبیت حالتهای فضای مساله، استفاده می کنند.

تابع ارزیابی (Evaluation Function): جستجوی آگاهانه یا هیوریستیک از تابع ارزیابی جهت هدایت روند جستجو، بهره می گیرد. تابع ارزیابی؛ حالت کلی تری از تابع هیوریستیک است. روش های جستجوی آگاهانه دارای استراتژی های معروف زیر است:

- جستجوی اول بهترین
 - جستجوی حریمانه
 - جستجوی A^*
- جستجوی حافظه محدود شده
 - روش IDA^*
 - روش MA^*
 - روش SMA^*
 - ...
- الگوریتم های اصلاح شونده تکراری
 - تپه نوردی
 - آنالینگ شبیه سازی شده
 - جستجوی پرتویی
 - الگوریتم ژنتیک
 - ..

جستجوی اول بهترین

پرسش: در درس طراحی الگوریتم‌ها، جستجوی اول بهترین، چگونه بود؟

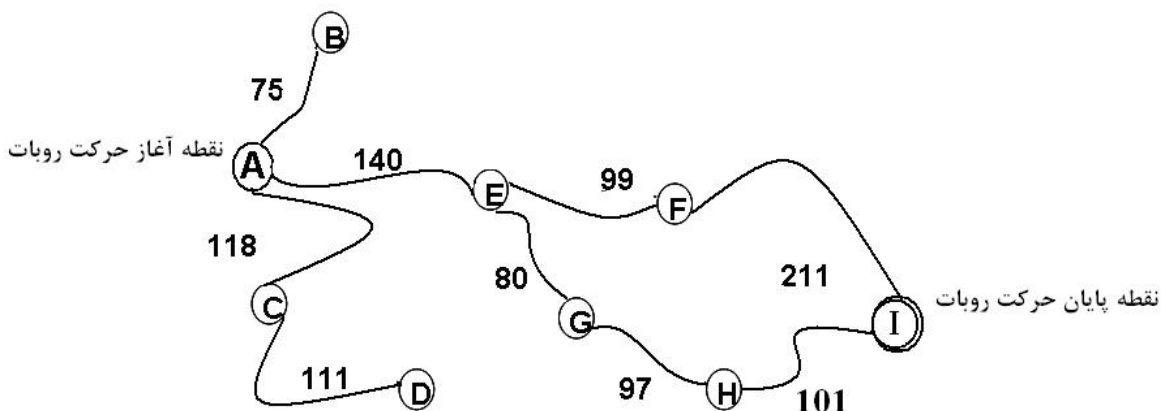
۱- جستجوی حریمانه (جستجوی اول بهترین حریمانه)

$h(n)$ = estimated cost of the cheapest path from the state at n to a goal state, $f(n) = h(n)$

ایده اصلی روش جستجوی اول بهترین حریمانه آن است که نخست گره‌ای گسترش یابد که به نظر می‌رسد، بهترین است (به هدف نزدیکتر است). برای ارزیابی و تعیین گره به نظر بهترین، از تابع هیوریستیک h استفاده می‌شود.

مثال: رباتی در نظر بگیرد که برای مسیریابی خود از نقطه A به نقطه I از روش جستجوی حریمانه در هر نقطه، برای انتخاب نقطه بعدی استفاده کند. فرض کنید که تابع هیوریستیک این ربات، فاصله خط مستقیم را توسط آنتن ماهواره‌ای بین دو نقطه محاسبه می‌شود. مسیر پیموده شده، توسط ربات چیست؟ مسافت پیموده شده چقدر است؟

State	$h(n)$
A	366
B	374
C	329
D	244
E	253
F	178
G	193
H	98
I	0



شکل ۳-۱: نقشه مسیریابی

$f(n) = h_{SLD}(n)$

مسافت مسیر پیموده شده توسط ربات، با روش جستجوی حریمانه: $A-E-F-I = 140 + 99 + 211 = 450$

پرسش: آیا مسیر بهتر (با طول کمتر) برای ربات وجود دارد؟ $A-E-G-H-I = 418$

نکته: بنابراین جستجوی حریمانه، بهینه نیست.

ویژگی‌های جستجوی حریمانه

- کامل نیست.
- بهینه نیست.
- پیچیدگی زمانی: $O(b^m)$
- پیچیدگی مکانی: $O(b^m)$
- کارایی روش به دقت و کارایی تابع هیوریستیک h بستگی دارد.

هیچ کس به خرد غایی نرسد، مگر آن را در خود جستجو کند. (ابن‌سین)

۲- روش جستجوی A*

الگوریتم A

- تابع ارزیابی $f(n) = g(n) + h(n)$ به کار می‌برد که تابع $g(n)$ کمترین هزینه از گره ریشه تا گره جاری n است. تابع $g(n)$ ، همان تابع هزینه در روش جستجوی سازگار با هزینه است و تابع $h(n)$ هم تابع هیوریستیکی است که در روش جستجوی حریمانه، توضیح داده شد.
- روش کامل نیست اگر تابع $h(n)$ بتواند نامتناهی شود.
- روش بهینه (پذیرفتنی) نیست.

الگوریتم A*

- الگوریتم A*، الگوریتم A ای است با تابع ارزیابی $f(n) = g(n) + h(n)$ و با محدودیت $h(n) \leq h^*(n)$.
- $h^*(n)$ هزینه واقعی در مسیر بهینه از گره جاری n تا گره هدف است و $h(n)$ هزینه تخمینی از گره جاری n تا گره هدف در مسیر بهینه است.
- $h(n) \leq h^*(n)$ پذیرفتنی (admissible) است اگر $h(n) \leq h^*(n)$.
- شرط و محدودیت admissible تضمینی است برای یافتن جواب بهینه در الگوریتم A*.
- الگوریتم A* کامل است اگر ضریب انشعاب متناهی و هر عملگر هزینه ثابت مثبت داشته باشد (اثبات می‌شود).
- الگوریتم A*، پذیرفتنی (admissible) است.
- الگوریتم A* بهینه است زیرا پذیرفتنی (admissible) است (اثبات می‌شود).

h must be an underestimate of the true optimal cost : $\forall n \quad h(n) \leq h^*(n)$

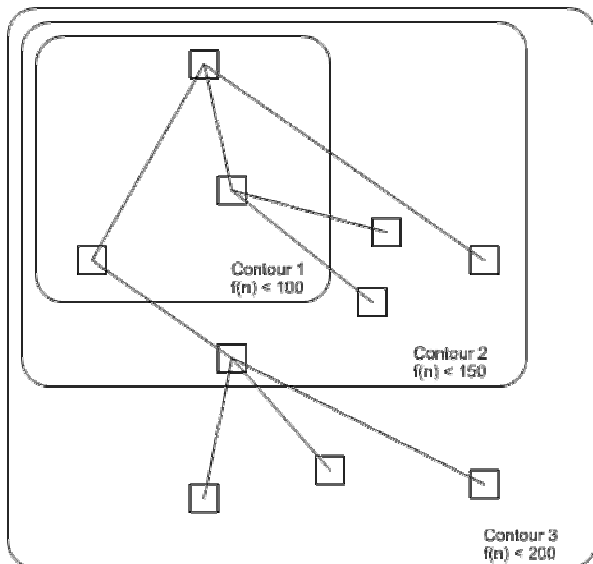
پرسش: پیچیدگی مکانی و زمانی الگوریتم A* چیست؟ مثال مسیریابی روبات قبلی را با روش جستجوی A* حل کنید.

مسیر راه حل: A-E-G-H-I = 418

پرسش: مشکل عمده روش جستجوی A* چیست؟ راه حل یا راه حل‌ها کدامند؟

جستجوی آگاهانه با حافظه محدود

Memory-conserving variations of A*: IDA*, MA*, SMA*



شکل ۳-۲: روند کلی جستجوی IDA*

۱- روش IDA*

پرسش: جستجوی IDS چگونه بود؟

Iterative Deepening A* = IDA*

مثال: روند کلی روش IDA*

Find all nodes

- Where $f(n) < 100$
- Don't expand any Where $f(n) > 100$
- Find all nodes
- Where $f(n) < 150$
- Don't expand any Where $f(n) > 150$

And so on...

ویژگی های روش IDA*

- کامل و بهینه است.
- پیچیدگی زمانی آن بستگی به تابع ارزیابی $f_cost(n)$ دارد.
- پیچیدگی مکانی آن خطی است زیرا پیاده سازی آن اساساً بر پایه روش IDS است.
- پاسخ روش IDA* همان پاسخ روش A* است

نکته: اگر اطلاعاتی تقریبی در مورد عمق جواب نداشته باشیم، روش جستجوی IDA* گزینه مناسبی است. چرا؟

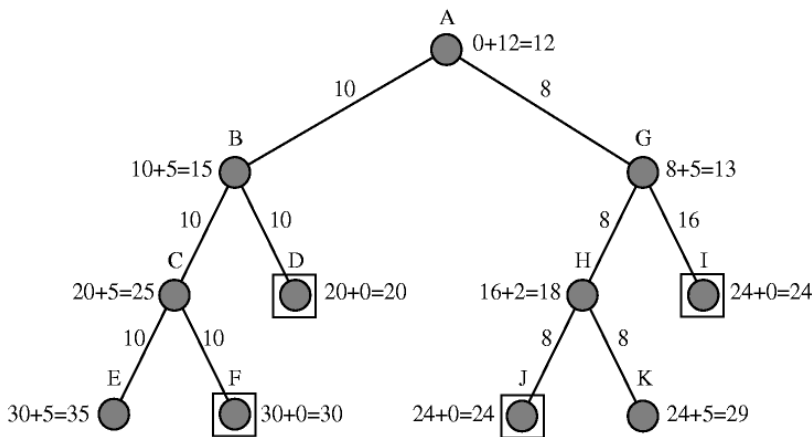
۲- روش MA*

جستجوی MA* نسخه پیشین SMA* است.

برای کسب اطلاعات بیشتر در مورد چگونگی عملکرد این الگوریتم، به مقاله [Chakrabarti 1989]، مراجعه کنید.

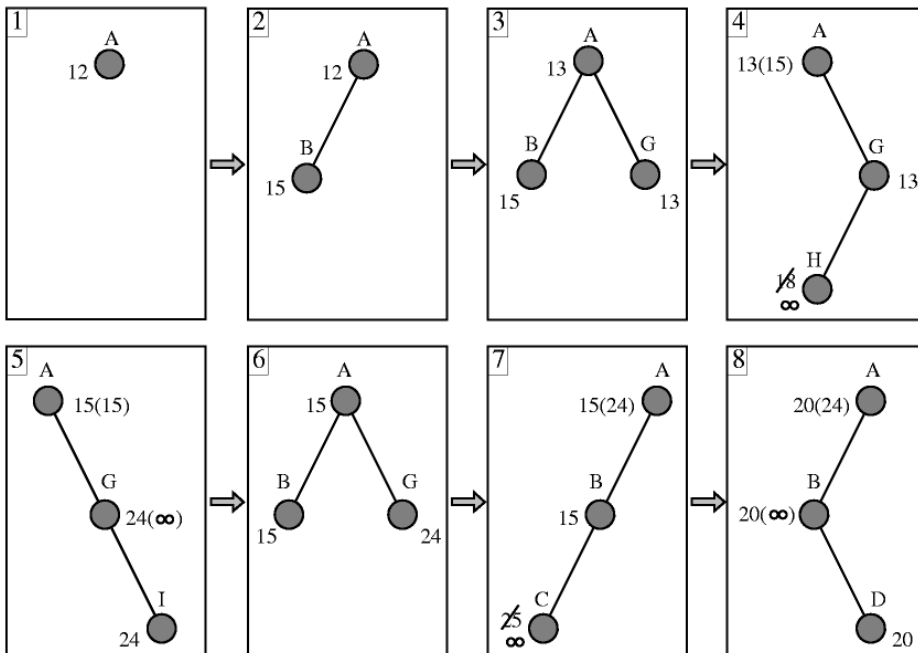
۳- روش SMA*

(Simplified Memory Bounded A*)



ویژگی های SMA*

- استفاده از تمام حافظه در دسترس
- جلوگیری از حالات تکراری تا جایی که حافظه امکان بدهد.
- کامل بودن آن بشرط آنکه حافظه برای ذخیره کم عمق ترین مسیرراه حل، موجود باشد.
- بهینه بودن آن بشرط آنکه حافظه برای ذخیره کم هزینه ترین مسیرراه حل، موجود باشد.
-



شکل ۳-۳: مراحل الگوریتم SMA* روی یک مثال

نکته: روش SMA* توسط استوارت

راسل در سال ۱۹۹۲ برای بهبود MA*،

ارائه شده است [Russell 1992].

پوشش: جستجوی اول بهترین بازگشتی یا RBFS روشی است که تلاش می کند تا جستجوی اول بهترین را با مصرف حافظه کم اجرا کند. چگونه؟

توابع هیوریستیک و طراحی آنها

$h(n)$ = Euclidean (straight-line) distance from n to goal.

مثال: از تابع هیوریستیک (تابع یافتاری) در مساله مسیریابی

۱- ساده سازی (Relaxation)

پرسش: مساله ساده یا راحت شده چیست؟

$h_1(n)$ = number of tiles out of place in n .

مثال: (معمای هشت)

$h_2(n)$ = total Manhattan distance (i.e., number of squares from desired location of each tile).

۲- ترکیب توابع یافتاری

اگر h_1 و h_2 توابع یافتاری پذیرفتنی باشند، آنگاه

$h_3 = \max(h_1, h_2)$ تابع یافتاری پذیرفتنی است. چرا؟

۳- اطلاعات آماری

- مساله مورد نظر با نمونه‌های مختلف و متفاوت، اجراء می‌شود.
- روی نتایج اجرای نمونه‌های مختلف تحلیل‌های آماری مثل میانگین، میانه و ... انجام می‌شود.

۴- استخراج ویژگی

بعضی مسائل دارای ویژگی‌هایی هستند که این ویژگی‌ها در محاسبه تابع یافتاری (هیوریستیک)، نقش مهمی دارند.

مثال: بازی شطرنج

- تعداد مهره‌ها
- تعداد حملات ممکن
- و ارزش مهره‌ها
- ...

نکته: برای در نظر گرفتن تمامی ویژگی‌ها، می‌توان از یک تابع ترکیب خطی استفاده کرد که ضرایب، وابسته به ارزش و میزان تاثیر ویژگی هاست.

پرسش: اگر دو تابع هیوریستیک داشته باشیم و به راحتی و از روی تعریف توابع هیوریستیک، نتوانیم تشخیص دهیم که کدامیک بهتر است، چه روشی یا معیاری برای تشخیص تابع هیوریستیک بهتر وجود دارد؟

ضریب انشعاب موثر

Effective Branching Factor = EBF(b^*)

$$N+1 = 1 + b^* + (b^*)^2 + \dots + (b^*)^d$$

$$N = 1 + b^* + (b^*)^2 + \dots + (b^*)^d$$

N : تعداد گره‌های درخت جستجو

d : عمق درخت جستجو

b^* : ضریب انشعاب موثر

پرسش: اگر جستجوی A^* در عمق ۵ با ۵۲ گره باشد، b^* آن چیست؟

در جدول ۱-۳، b^* سه روش برای معمای هشت، نشان داده شده است.

- h_1 = number of misplaced tiles
- h_2 = sum of distances of tiles to their goal positions

جدول ۱-۳

d	IDS	A_1^*	A_2^*
2	2.45	1.79	1.79
6	2.73	1.34	1.30
12	2.78 (3,644,035)	1.42 (227)	1.24 (73)
16	--	1.45	1.25
20	--	1.47	1.27
24	--	1.48 (39,135)	1.26 (1,641)

الگوریتم‌های جستجوی محلی یا الگوریتم‌های بهبودشونده تکراری

- الگوریتم تپه‌نوردی
- آنالینگ شبیه‌سازی شده
- جستجوی پرتویی محلی
- الگوریتم ژنتیک
- ...

نکته: این گونه روش‌ها، برای حل مسایل بهینه‌سازی، کاربرد فراوانی دارند. چرا؟



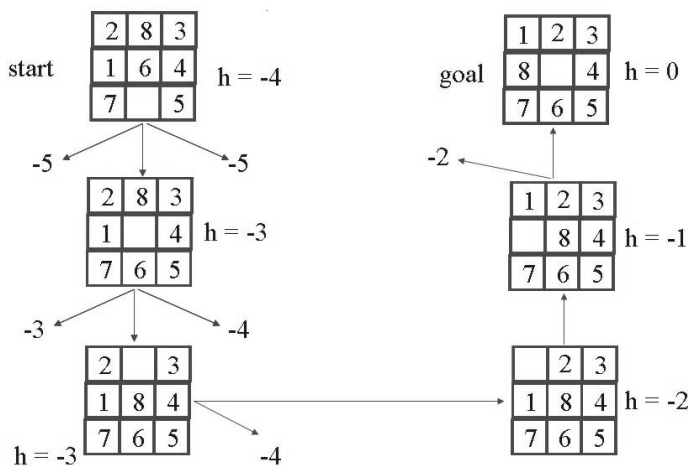
شکل ۳-۴: جستجوی تپه‌نوردی

تپه‌نوردی (Hill Climbing)

ایده یا شعار اصلی روش تپه‌نوردی:

Always choose the next best successor state.
Stop when no improvement possible

مثال: معمای هشت را با روش تپه‌نوردی حل کنید.



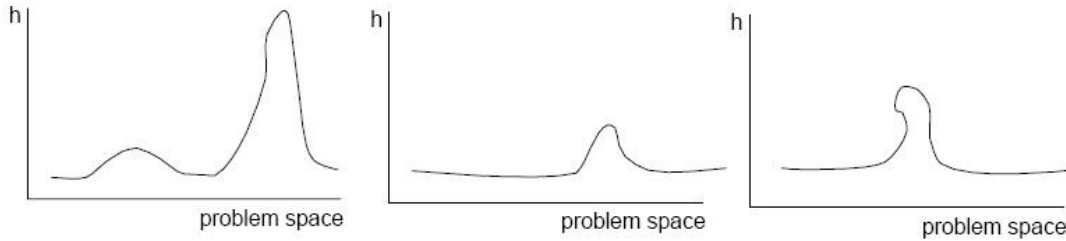
$$f(n) = -(\text{number of tiles out of place})$$

شکل ۳-۵: حل معمای هشت با روش تپه‌نوردی

بردن به این معنی نیست که تو همیشه اول باشی، بردن به این معنی است که تو بهتر از قبلت باشی. (بونی بلایر)

روش تپهنوردی به دلیل استراتژی ساده‌ای که در گزینش حالت‌های بعدی دارد، ممکن است دچار مشکلات زیر شود:

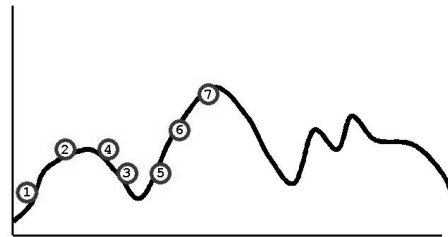
- ماکزیم‌های محلی (Local Maxima)
- فلات‌ها (Plateaus)
- تیغه‌ها (Ridges)



شکل ۳-۶: مشکلات روش تپهنوردی

پرسش: چه راه‌هایی برای مشکلات روش تپهنوردی، پیشنهاد می‌کنید؟

آنالینگ شبیه‌سازی شده (Simulated Annealing)



شکل ۳-۷: روند کلی آنالینگ شبیه‌سازی شده

جستجوی پرتویی (Beam Search)

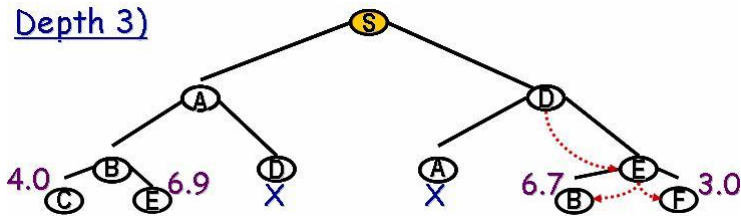
این روش، گسترش و توسعه یافته روش تپهنوردی است. مراحل جستجوی تپهنوردی روی یک مثال در شکل ۳-۸، در چهار گام نشان داده شده است.

گام‌های جستجوی پرتوی محلی

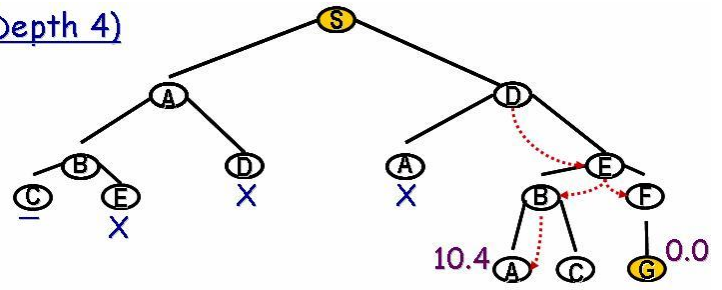
- ۱- شروع جستجو با k حالت تصادفی
 - ۲- ایجاد همه حالت‌های بعدی k حالت
 - ۳- نگهداری k تا بهترین از حالت‌های ایجاد شده
 - ۴- بازگشت به گام دوم
- بهبود الگوریتم

Optimization: ignore leaves that are not goal nodes (see C)

Depth 3



Depth 4



شکل ۳-۸: مراحل الگوریتم جستجوی پرتویی با $k=2$

برای پرش‌های بلند، گاهی نیاز است چند گامی پس رویم. (ارد بزرگ)

پرسش: چه ارتباطی از نظر مفهومی بین تپه‌نوردی و جستجوی پرتویی است؟

نکته: در بعضی از کتب و مقالات برای جستجوی پرتویی، به جای $f(n) = h(n)$ از $f(n) = g(n) + h(n)$ استفاده شده‌است.

ویژگی‌های جستجوی پرتویی

- کامل نیست. چرا؟
- بهینه نیست. چرا؟

جستجوی Online

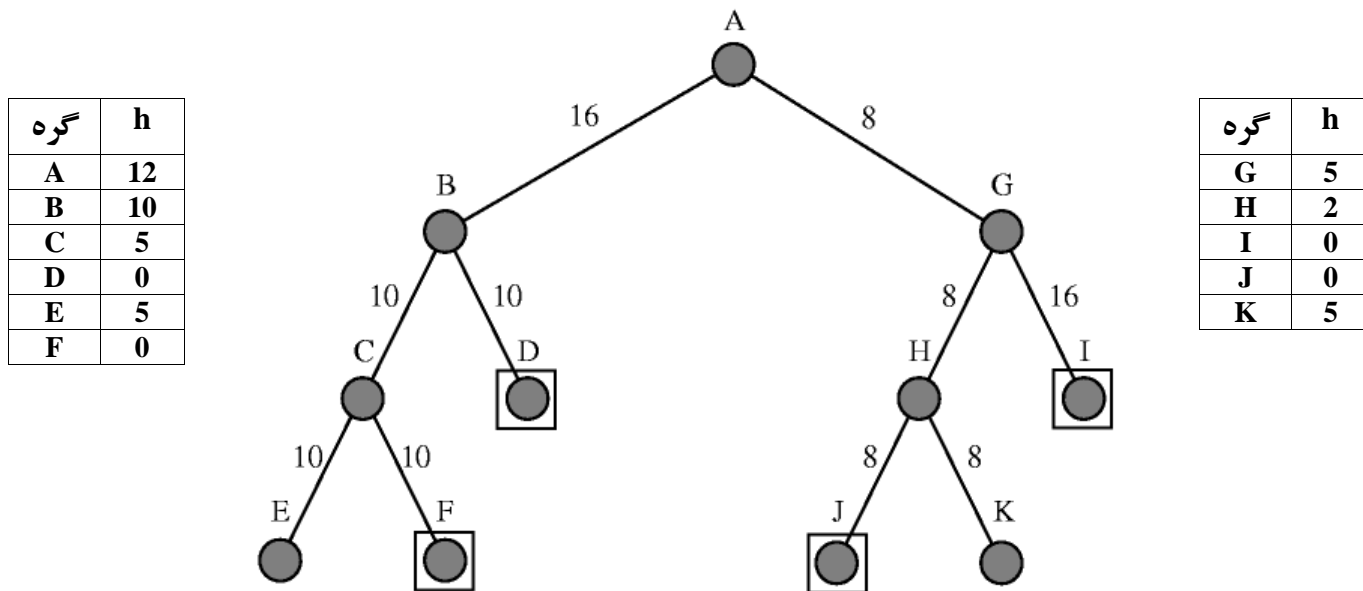
جستجوی آنلاین برای فضاهای حالت با اطلاعات ناقص (یا عدم اطلاعات) مفید است. عامل‌ها در مسائل واقعی، معمولاً اطلاعات کاملی از محیط ندارند.

- الگوریتم تپه‌نوردی
- جستجوی $LRTA^*$

تمرین‌ها

تمرین ۱-۳: با توجه درخت جستجوی شکل ۳-۹ و مقادیر تابع هیوریستیک برای گره‌ها، در جداول زیر، روش‌های جستجوی زیر را بر این درخت جستجو اعمال کرده و حاصل را برای هر روش، گام به گام نشان دهید.

الف- روش SMA^* ، ب- جستجوی حریصانه (اول بهترین حریصانه)، ج- جستجوی تپه‌نوردی و د- جستجوی پرتویی (محدودودیت حافظه برای روش SMA^* ، ۳ گره فرض شود و برای جستجوی پرتویی $k=2$ در نظر گرفته شود)



شکل ۳-۹: درخت جستجو

تمرین ۲-۳: جستجوی RBFS را با جستجوی $LRTA^*$ و جستجوی $ITSA^*$ مقایسه کنید. مزایا و معایب هر روش کدامند؟

تمرین ۳-۳: برای هر یک از روش‌های جستجو معرفی شده، کاربرد یا کاربردهایی واقعی مثال بزنید.

تمرین ۴-۳: روش‌هایی نظیر A^* که بررسی شد، روی درخت جستجو عمل می‌کردند. چگونه این گونه روش‌ها، برای گراف جستجو می‌تواند استفاده شود؟

تمرین ۵-۳: تحقیق کنید جستجوی MA^* و جستجوی $GSMA^*$ چه شباهت‌ها و تفاوت‌هایی با SMA^* دارند؟

تاریک‌ترین ساعت شب، درست ساعات قبل از طلوع خورشید است.