حل مسئله با استفاده از جستجو
(جستجوی ناآگاهانه)

دانشگاه صنعتی قوچان

محمدحسین سیگاری

فهرست

- تعریف مسئله و فرموله کردن آن
- جستجوی درخت
  - BFS، UCS، DFS، DLS، IDS و BDS
- جستجوی گراف

مراحل کلی حل مسئله

- فرموله کردن هدف (Goal Formulation)
  - وضعیت های مطلوب نهایی کدامند؟
- فرموله کردن مسئله (Problem Formulation)
  - چه فعالیت ها و چه وضعیت هایی برای رسیدن به هدف موجود است؟
- جستجو (Search)
  - تعیین دنباله هایی از فعالیت ها که منجر به حالاتی با مقدار شناخته شده می شود و انتخاب بهترین دنباله از دنباله فعالیت ها برای رسیدن به هدف
- اجرا (Execute)
  - وقتی دنباله فعالیت مطلوب پیدا شد، فعالیت های پیشنهادی آن اجرا می شود

فرموله کردن مسئله و هدف

حالت اولیه (Initial States)

حالت هایی که عامل می تواند حرکت خود را از آنجا شروع کند

تابع جانشین (Successor Function)

تعیین فعالیت های مجاز در هر حالت و حالت جدیدی که با انجام آن فعالیت ها حاصل می گردد

آزمون هدف (Goal Test)

تعیین می کند که آیا حالت کنونی، حالت هدف است یا خیر

هزینه مسیر (Path Cost)

یک تابع جمع شونده (Additive) از هزینه هر مرحله

تابع جانشین

- در هر state تعیین می کند که actionهای مجاز چیست و با انجام هر action، عامل به چه state جدیدی می رود
- مجموعه ای از زوج مرتب (action,state)

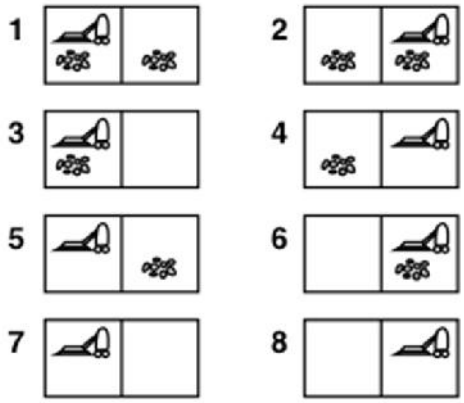
$$S(x_i) = \{(a_j, x_j)\}$$

- با دانستن حالت های اولیه و تابع جانشین، فضای حالت قابل محاسبه است

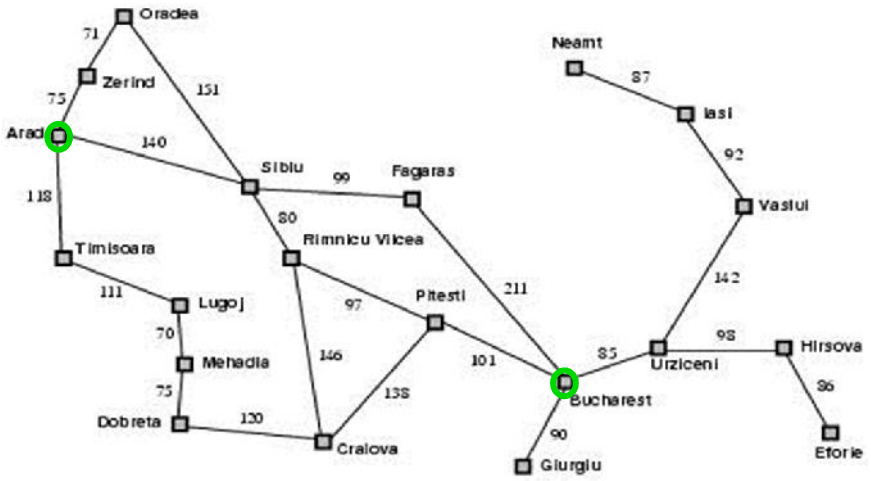
هزینه مسیر

- هزینه مسیر (Path Cost Function)
  - مجموع هزینه ای که عامل پرداخت کرده تا از یک حالت اولیه به حالت فعلی رسیده است
- هزینه هر مرحله (Step Cost Function)
  - با انجام هر action عامل یک هزینه پرداخت کرده و از حالتی به حالت دیگر می رود
  - C(x,a,y)

مثال: جهان جاروبرقی



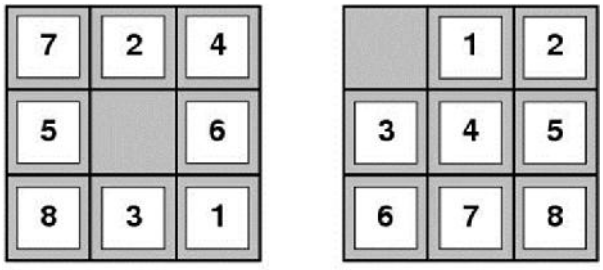
مثال: سفر در کشور رومانی



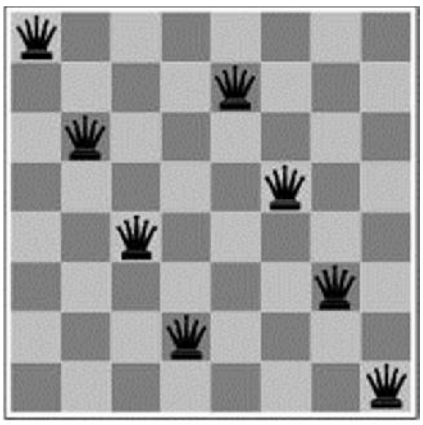
مثال: سفر در کشور رومانی

- صورت مسأله: رفتن از آراد به بخارست
- فرموله کردن هدف: رسیدن به بخارست
- فرموله کردن مسئله:
  - وضعیتها: شهرهای مختلف
  - فعالیتها: حرکت بین شهرها
- جستجو: دنباله ای از شهرها مثل:آراد، سیبیو، فاگارس، بخارست
  - این جستجو با توجه به کم هزینه ترین مسیر انتخاب می شود

مثال: معمای ۸ (8-Puzzle)



مسئله ۸ وزیر (8 Queens)



1- Incremental
2- Complete

تجرید/انتزاع اطلاعات (Abstraction)

- دنیای واقعی بسیار پیچیده است و برای کاربرد در ماشین هوشمند، باید اطلاعات را به صورت مجرد و فشرده استفاده نمود
- تجرید اطلاعات برای ساده سازی فرموله کردن مسئله و حل آن می باشد
- تجرید/انتزاع: حذف اطلاعات غیرلازم تاحدی که باعث کاهش پیچیدگی و افزایش کارایی گردد
- (Abstract) state = set of real states
- (Abstract) action = complex combination of real actions.
- (Abstract) solution = set of real paths that are solutions in the real world

تجرید/انتزاع اطلاعات (Abstraction)

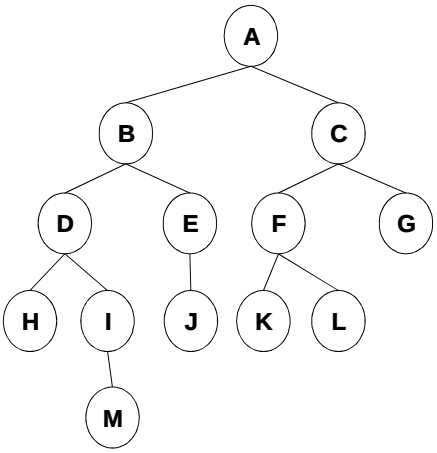
- تبدیل مسئله به گراف جهت دار یا درخت

مسئله	گراف	درخت
حالت (State)	گره (Node)	گره (Node)
فعالیت (Action)	لبه (Edge)	لبه (Edge)
فضای حالت (State Space)	گراف جهت دار (Directed Graph)	درخت (Tree)

فهرست

- تعریف مسئله و فرموله کردن آن
- جستجوی درخت
  - BFS، UCS، DFS، DLS، IDS و BDS
- جستجوی گراف

جستجوی درخت



- انواع گره:
  - ریشه (Root)
  - برگ (Leaf)
  - والد (Parent)
  - فرزند (Child)
  - والدبزرگ (Grandparent)
  - نوه (Grandchild)
  - جد (Predecessor)
  - جانشین (Successor)
  - همزاد (Sibling)
  - حاشیه (Fringe)
- گره ای که تولید شده، اما هنوز مورد آزمون هدف قرار نگرفته یا بسط داده نشده است

بررسی کارایی الگوریتم جستجو

- کامل بودن (Completeness)
  - آیا الگوریتم تضمین می کند که در صورت وجود راه حل، آن را بیابد؟
- بهینگی (Optimality)
  - آیا این راهبرد، راه حل بهینه ای را ارائه می کند؟
- پیچیدگی زمانی (Time Complexity)
  - چقدر طول میکشد تا راه حل را پیدا کند؟
  - معمولا بر اساس *تعداد گره های تولید شده* برای جستجو
- پیچیدگی فضا (Space Complexity)
  - برای جستجو چقدر حافظه نیاز دارد؟
  - معمولا بر اساس *حداکثر تعداد گره های ذخیره شده در حافظه*

بررسی کارایی الگوریتم جستجو

- پارامترهای مورد استفاده برای بررسی کارایی الگوریتم جستجو

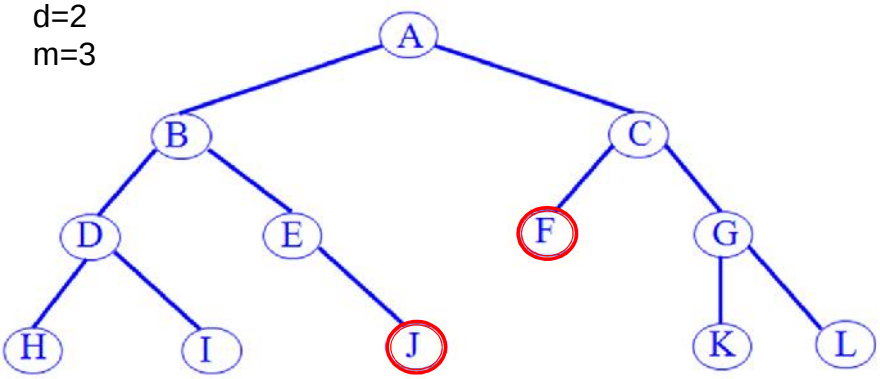
- b
  - حداکثر تعداد انشعابات یک گره (branch factor)

- d
  - عمق هدفی که کمترین هزینه را دارد

- m
  - حداکثر عمق فضای جستجو (ممکن است بی نهایت باشد)

جستجوی درخت

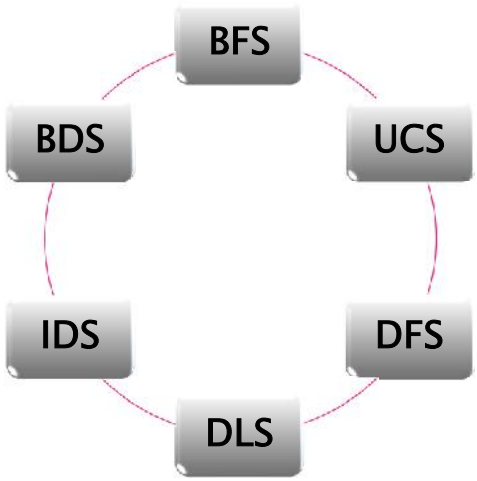
b=2
d=2
m=3



جستجوی کور (ناآگاهانه)

- کور (Blind) = ناآگاهانه (Uninformed)
- علت ناآگاهی این است که الگوریتم هیچ اطلاعاتی غیر از تعریف مسئله در اختیار ندارد

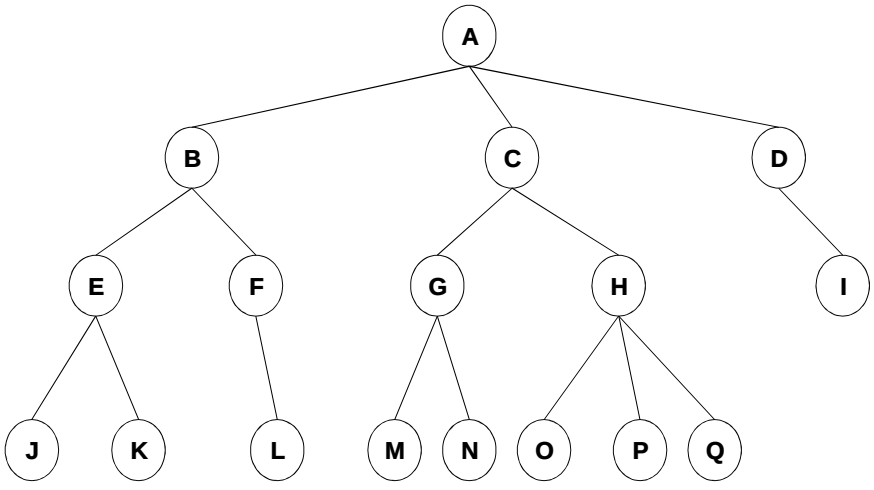
انواع جستجوی ناآگاهانه



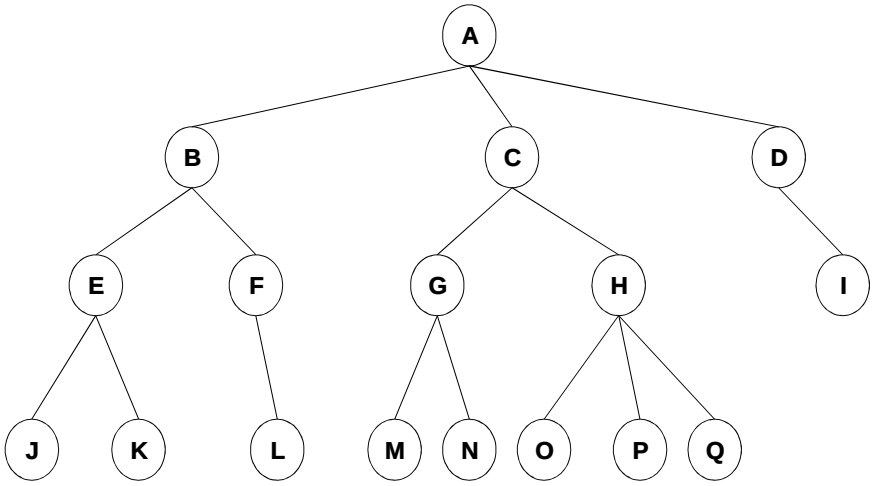
انواع جستجوی ناآگاهانه

- BFS: جستجوی اول سطح (Breadth First Search)
- UCS: جستجوی هزینه یکنواخت (Uniform Cost Search)
- DFS: جستجوی اول عمق (Depth First Search)
- DLS: جستجوی عمق محدود (Depth Limited Search)
- IDS: جستجوی عمیق کننده تکراری (Iterative Deepening Search)
- BDS: جستجوی دو طرفه (Bidirectional Search)

اجرای انواع جستجوی



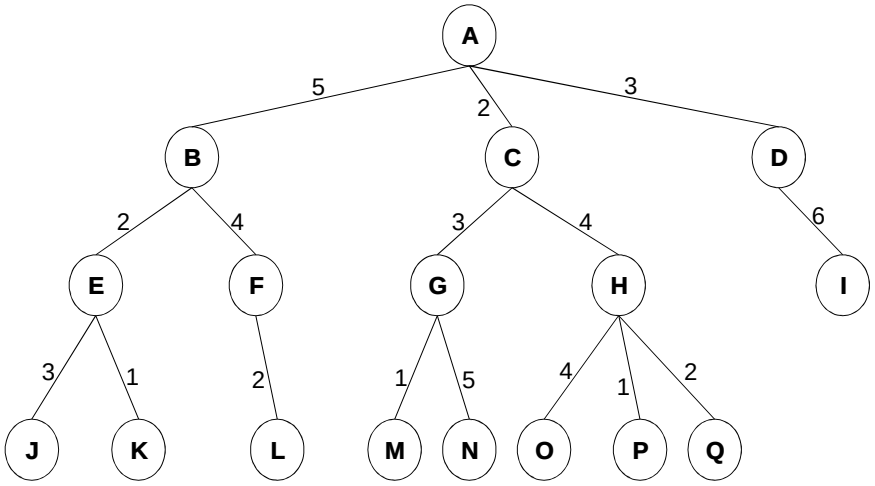
جستجوی اول سطح (BFS)



جستجوی اول سطح (BFS)

- کامل بودن:
  - کامل است، اگر هزینه مسیر غیرنزولی باشد. به عبارت دیگر هزینه هر یال نامنفی باشد
  - اگر هزینه مسیر نزولی باشد، هدف در عمق بی نهایت قرار دارد
- بهینه بودن:
  - بهینه است، اگر هزینه تمام یال ها یکسان باشد
- پیچیدگی زمانی:
  - $O(b^{d+1})$
- پیچیدگی فضا:
  - $O(b^{d+1})$

جستجوی هزینه یکنواخت (UCS)

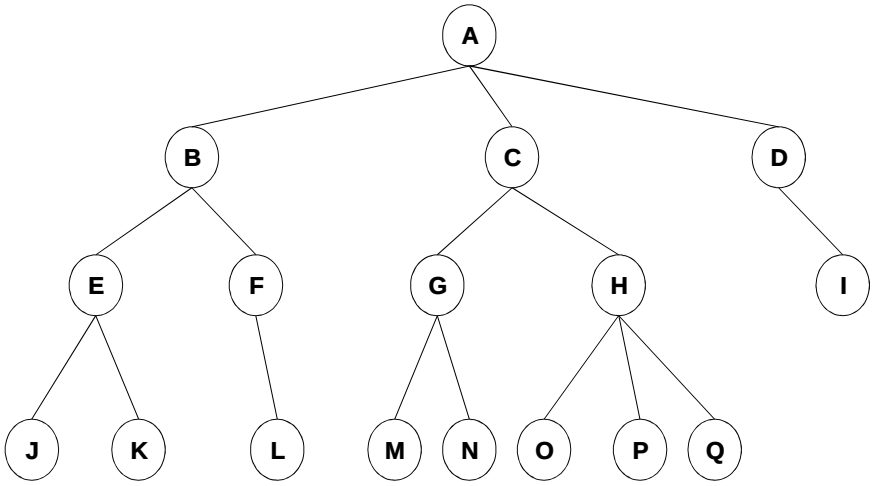


جستجوی هزینه یکنواخت (UCS)

- کامل بودن:
  - کامل است، اگر هزینه مسیر غیرنزولی باشد. به عبارت دیگر هزینه هر یال نامنفی باشد
  - اگر هزینه مسیر نزولی باشد، هدف در عمق بی نهایت قرار دارد
- بهینه بودن:
  - بهینه است
- پیچیدگی زمانی:
  - $O(b^{[C^*/e]})$ که e کمترین هزینه هر یال و C^* هزینه بهترین هدف است
- پیچیدگی فضا:
  - $O(b^{[C^*/e]})$ که e کمترین هزینه هر یال و C^* هزینه بهترین هدف است

BFS is a special type of UCS

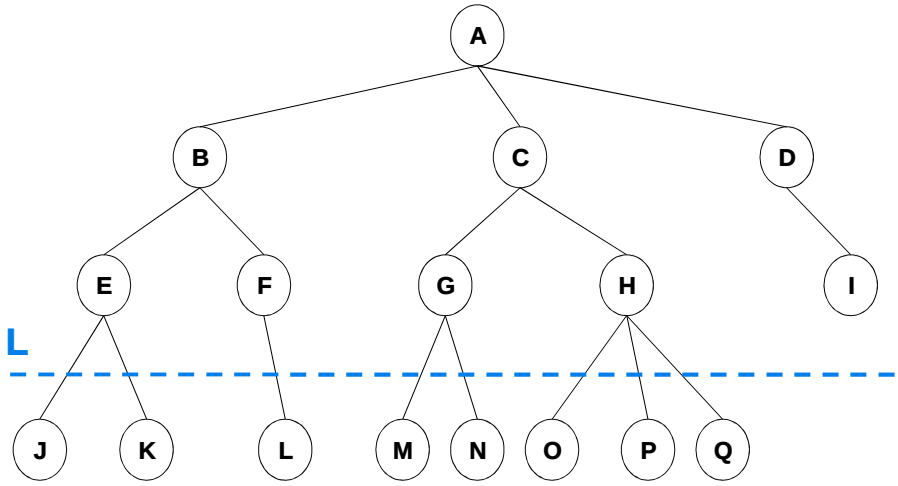
جستجوی اول عمق (DFS)



جستجوی اول عمق (DFS)

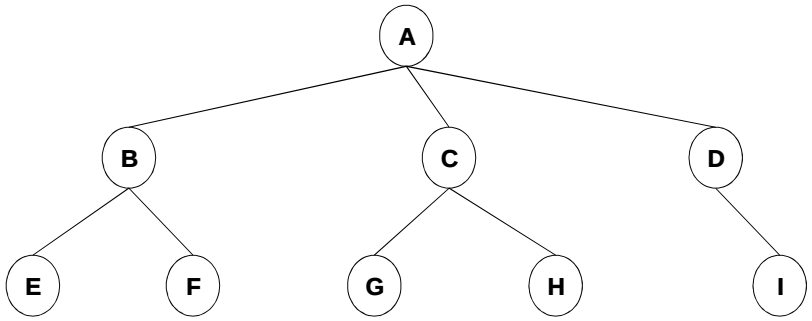
- کامل بودن:
  - کامل است، اگر حداکثر عمق درخت (m) بی نهایت نباشد
- بهینه بودن:
  - بهینه نیست
- پیچیدگی زمانی:
  - $O(b^m)$
- پیچیدگی فضا:
  - $O(bm)$

جستجوی عمق محدود (DLS)



هوش مصنوعی و سیستم خبره ۲۹

جستجوی عمق محدود (DLS)

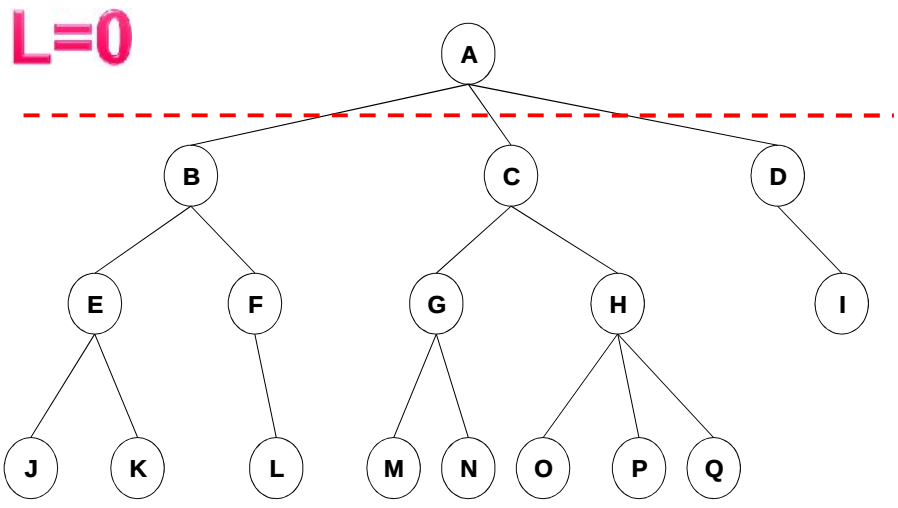


هوش مصنوعی و سیستم خبره ۳۰

جستجوی عمق محدود (DLS)

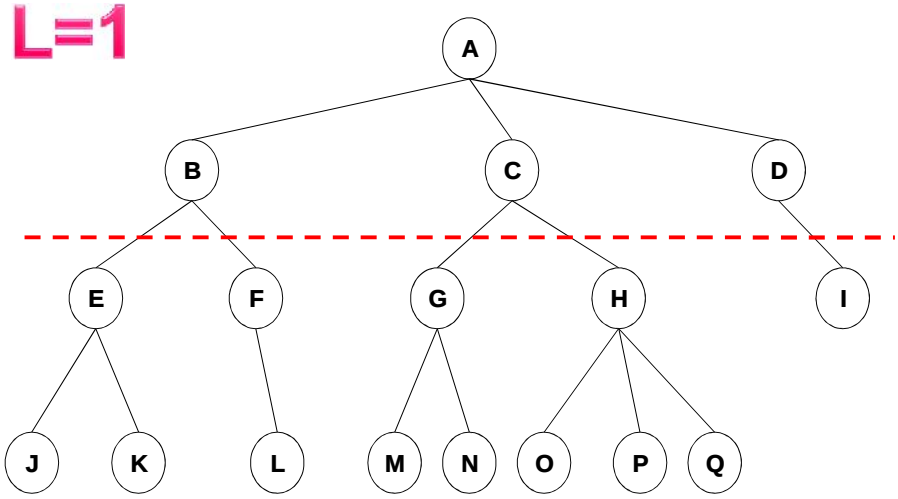
- کامل بودن:
  - کامل است، اگر $L \geq d$
- بهینه بودن:
  - بهینه است اگر $L = d$
- پیچیدگی زمانی:
  - $O(b^L)$
- پیچیدگی فضا:
  - $O(bL)$

جستجوی عمیق کننده تکراری (IDS)



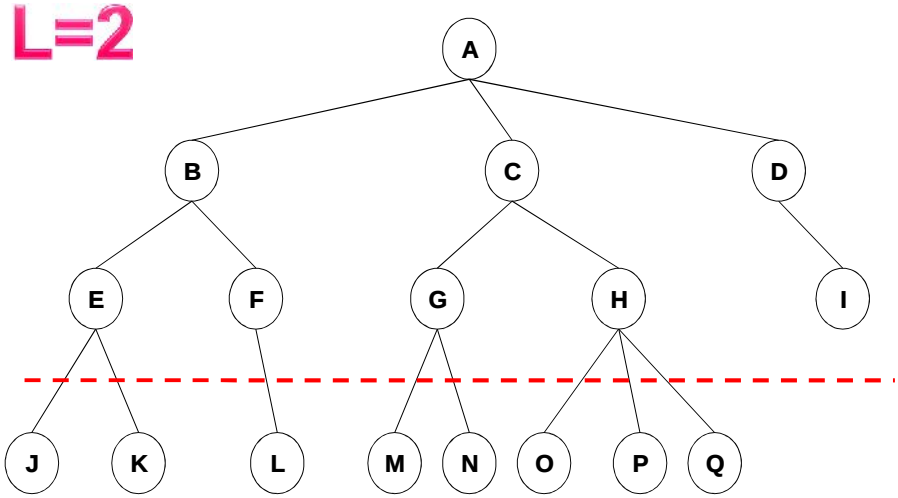
جستجوی عمیق کننده تکراری (IDS)

L=1

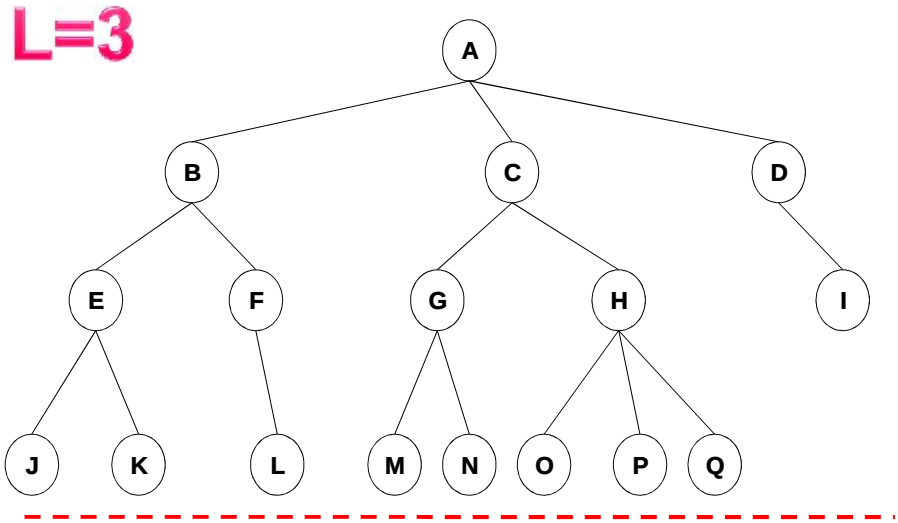


جستجوی عمیق کننده تکراری (IDS)

L=2



جستجوی عمیق کننده تکراری (IDS)



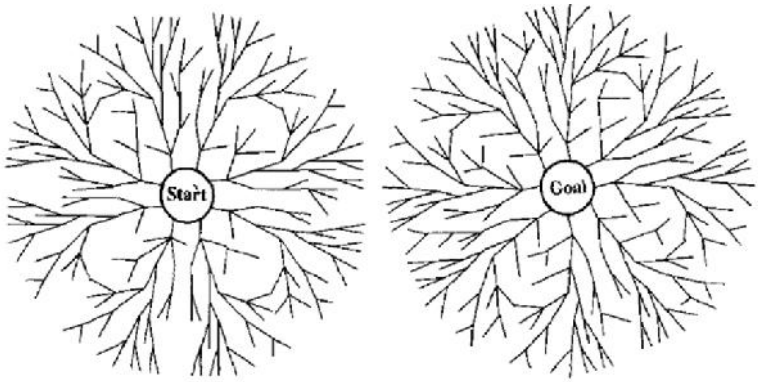
جستجوی عمیق کننده تکراری (IDS)

- کامل بودن:
  - کامل است، اگر هزینه مسیر غیرنزولی باشد. به عبارت دیگر هزینه هر یال نامنفی باشد
  - اگر هزینه مسیر نزولی باشد، هدف در عمق بی نهایت قرار دارد
- بهینه بودن:
  - بهینه است، اگر هزینه تمام یال ها یکسان باشد
- پیچیدگی زمانی:
  - $O(b^{d+1})$
- پیچیدگی فضا:
  - $O(bd)$

جستجوی دو طرفه (BDS)

- اگر بتوان علاوه بر حرکت از حالت اولیه به سمت هدف، از هدف نیز به سمت حالت اولیه حرکت کنیم، می توان از جستجوی دو طرفه استفاده نمود.
- در این حالت باید عملگرهای مسئله معکوس پذیر باشند. یعنی علاوه بر داشتن تابع Successor هر حالت، باید تابع Predecessor آن را هم داشته باشیم.
- مزیت:
  - کاهش حجم محاسبات
  - امکان جستجو با استفاده از سیستم های توزیع شده

جستجوی دو طرفه (BDS)



جستجوی دو طرفه (BDS)

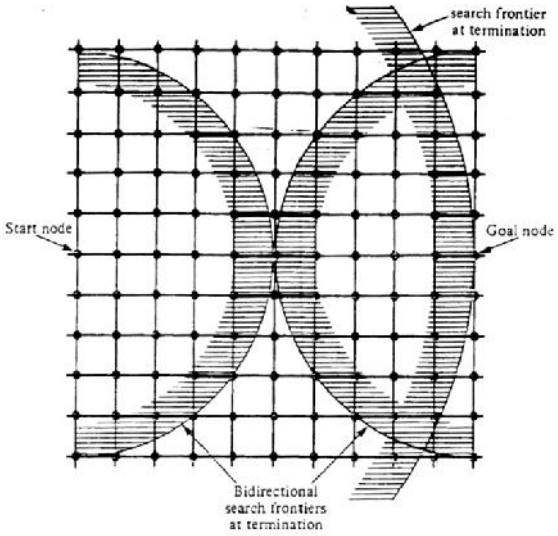


Fig. 2.10 Bidirectional and unidirectional breadth-first searches.

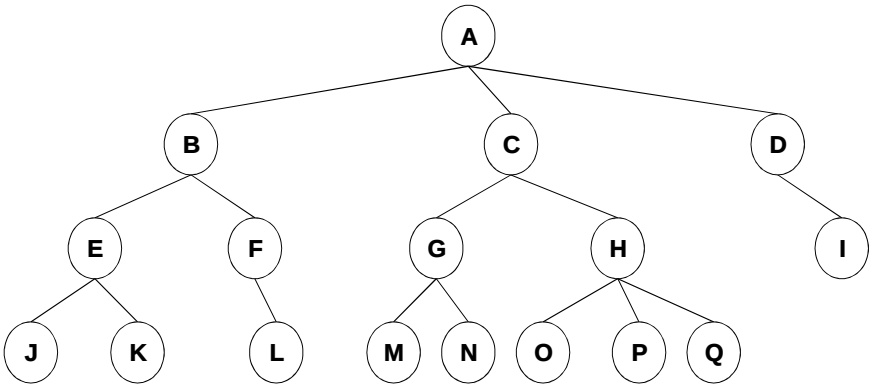
جستجوی دو طرفه (BDS)

- کامل بودن:
  - کامل است، اگر هزینه مسیر غیرنزولی باشد. به عبارت دیگر هزینه هر یال نامنفی باشد
  - اگر هزینه مسیر نزولی باشد، هدف در عمق بی نهایت قرار دارد
- بهینه بودن:
  - بهینه است، اگر هزینه تمام یال ها یکسان باشد و از جستجوی اول سطح برای جستجو از طرفین استفاده گردد
- پیچیدگی زمانی:
  - $O(b^{d/2+1})$
- پیچیدگی فضا:
  - $O(b^{d/2+1})$

نکاتی در جستجو

- خروجی نهایی یک جستجو شامل دو رشته توالی گره هاست:
  - گره های تولید شده (Generated)
  - گره های تست شده (Tested)
- ترتیب گره ها، روند جستجو را مشخص می کند.
- برای تولید گره های هم ارز، گره ای زودتر تولید شود که در سمت چپ قرار دارد
- گره ای که زودتر تولید شده، زودتر تست شود.
- آخرین گره ای که تست می شود، گره هدف است. بعد از آن جستجو متوقف می گردد

مثالی از نحوه تولید و تست گره ها در DFS



Gen: [A], [B, C, D], [E, F], [J, K], [L], [G, H], [M, N], [O, P, Q], [I]
Test: A, B, E, J, K, F, L, C, G, M, N, H, O, P, Q, D, I

فهرست

- تعریف مسئله و فرموله کردن آن
- جستجوی درخت
  - BFS، UCS، DFS، DLS، IDS و BDS
- جستجوی گراف

جستجوی گراف

- مهمترین مشکل در جستجوی گراف: وجود حلقه
- دو رویکرد به حل مسئله جستجوی گراف
  - تبدیل گراف به درخت و جستجوی درخت حاصل
  - جستجوی مستقیم گراف و توجه به حلقه ها
- در جستجوی گراف، با توجه به نوع جستجوی گراف، هر گره ممکن است چند بار تولید یا تست شوند.

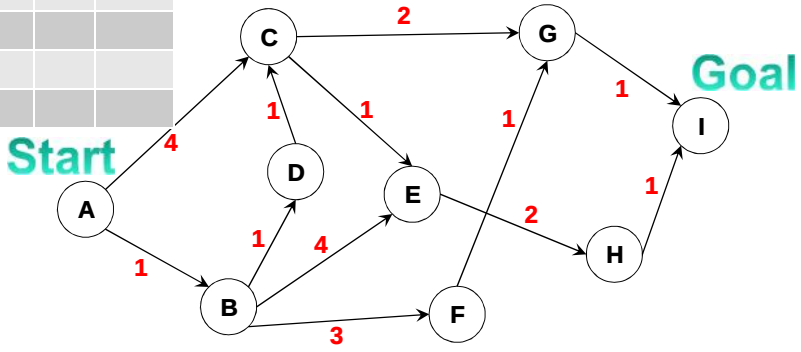
جستجوی گراف با روش جستجوی مستقیم گراف

- در این روش جستجو مستقیما بر روی گراف انجام می شود
- از یک جدول استفاده می شود که شامل دو بخش است
  - مشاهده یا عدم مشاهده شدن گره (Visited)
  - مسیر بهینه از گره شروع تا گره (Best Path)
    - هزینه مسیر بهینه از گره شروع تا گره
- یک گره فقط یک بار تولید می گردد، ولی ممکن است در مسیرهای مختلف از گراف چندین بار تست شود.
- در گراف جهت دار به جهت مسیر توجه شود

Node	Visited	Cost	Best Parent
A			
B			
C			
D			
E			
F			
G			
H			
I			

جستجوی گراف با روش UCS

- جستجوی مستقیم گراف



Node	Visited	Cost	Best Parent
A		0	–
B			
C			
D			
E			
F			
G			
H			
I			

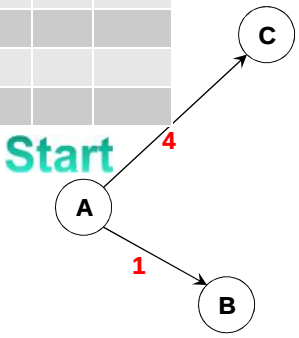
Start

A

جستجوی گراف با روش UCS

Gen: [A]
Test: -

Node	Visited	Cost	Best Parent
A	x	0	–
B		1	A
C		4	A
D			
E			
F			
G			
H			
I			



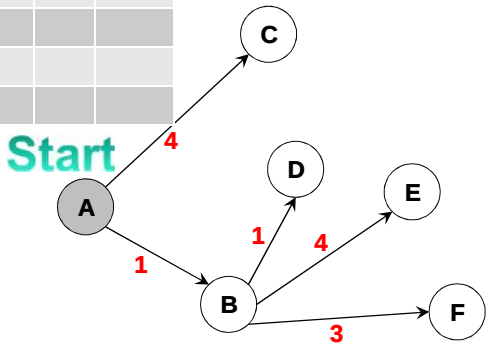
جستجوی گراف با روش UCS

Gen: [A], [B,C]
Test: A

Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C		4	A
D		2	B
E		5	B
F		4	B
G			
H			
I			

جستجوی گراف با روش UCS

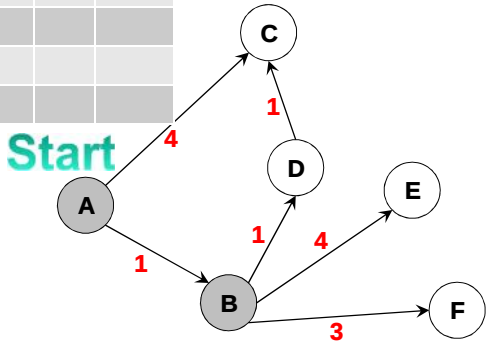
Gen: [A], [B,C], [F,E,D]
Test: A, B



Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C		3	D
D	x	2	B
E		5	B
F		4	B
G			
H			
I			

جستجوی گراف با روش UCS

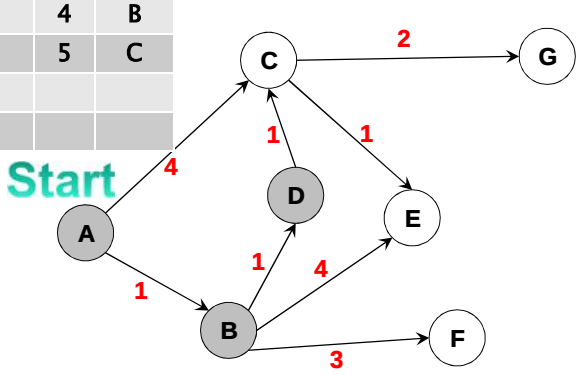
Gen: [A], [B,C], [F,E,D]
Test: A, B, D



Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C	x	3	D
D	x	2	B
E		4	C
F		4	B
G		5	C
H			
I			

جستجوی گراف با روش UCS

Gen: [A], [B,C], [F,E,D], [G]
Test: A, B, D, C

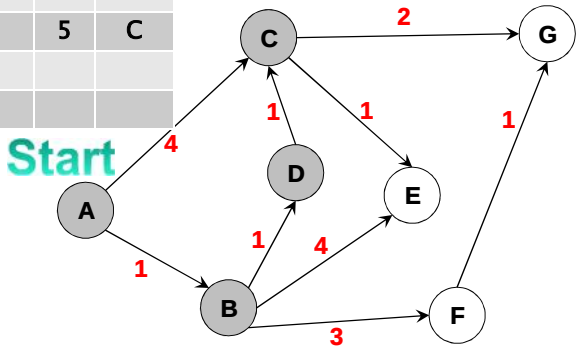


هوش مصنوعی و سیستم خبره ۵۱

Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C	x	3	D
D	x	2	B
E		4	C
F	x	4	B
G		5	C
H			
I			

جستجوی گراف با روش UCS

Gen: [A], [B,C], [F,E,D], [G]
Test: A, B, D, C, F

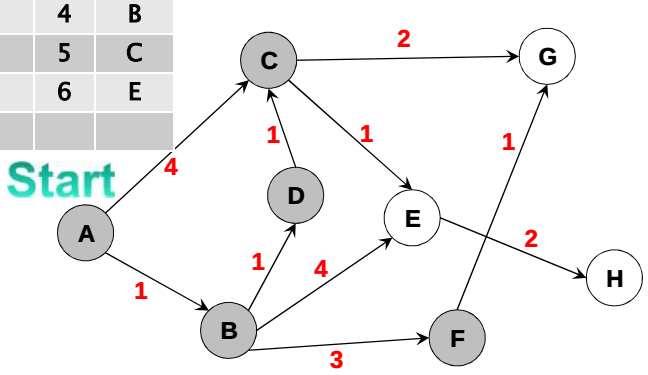


هوش مصنوعی و سیستم خبره ۵۲

Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C	x	3	D
D	x	2	B
E	x	4	C
F	x	4	B
G		5	C
H		6	E
I			

جستجوی گراف با روش UCS

Gen: [A], [B,C], [F,E,D], [G], [H]
Test: A, B, D, C, F, E

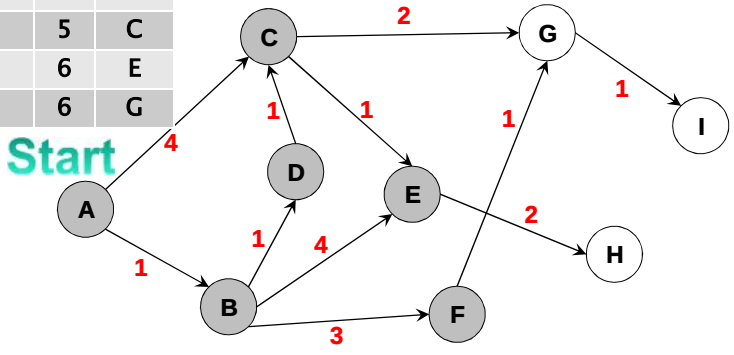


هوش مصنوعی و سیستم خبره ۵۳

Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C	x	3	D
D	x	2	B
E	x	4	C
F	x	4	B
G	x	5	C
H		6	E
I		6	G

جستجوی گراف با روش UCS

Gen: [A], [B,C], [F,E,D], [G], [H], [I]
Test: A, B, D, C, F, E, G

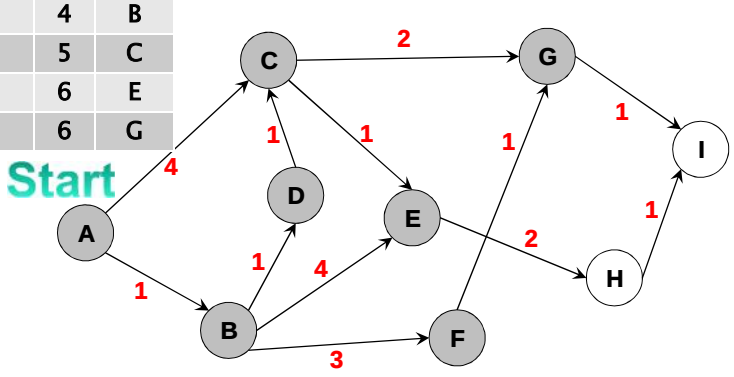


هوش مصنوعی و سیستم خبره ۵۴

Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C	x	3	D
D	x	2	B
E	x	4	C
F	x	4	B
G	x	5	C
H	x	6	E
I		6	G

جستجوی گراف با روش UCS

Gen: [A], [B,C], [F,E,D], [G], [H], [I]
Test: A, B, D, C, F, E, G, H

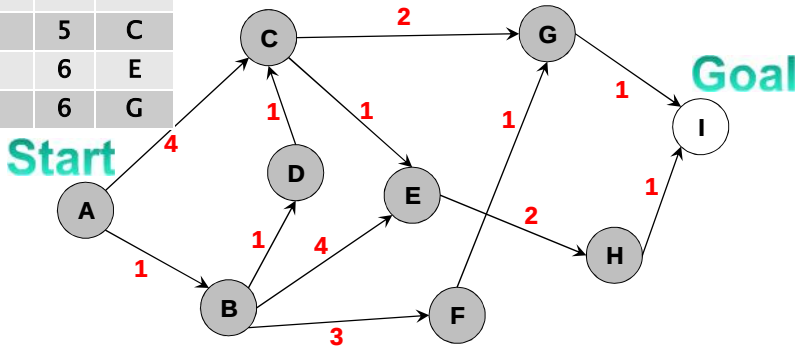


هوش مصنوعی و سیستم خبره ۵۵

Node	Visited	Cost	Best Parent
A	x	0	–
B	x	1	A
C	x	3	D
D	x	2	B
E	x	4	C
F	x	4	B
G	x	5	C
H	x	6	E
I	x	6	G

جستجوی گراف با روش UCS

Gen: [A], [B,C], [F,E,D], [G], [H], [I]
Test: A, B, D, C, F, E, G, H, I



هوش مصنوعی و سیستم خبره ۵۶