بسم الله الرحمن الرحیم

دانشگاه فنی وحرفه ای

هوش مصنوعی(فصل سوم)

میرمحمود ابوالحسنی

جستجوهای ناآگاهانه

فهرست مطالب

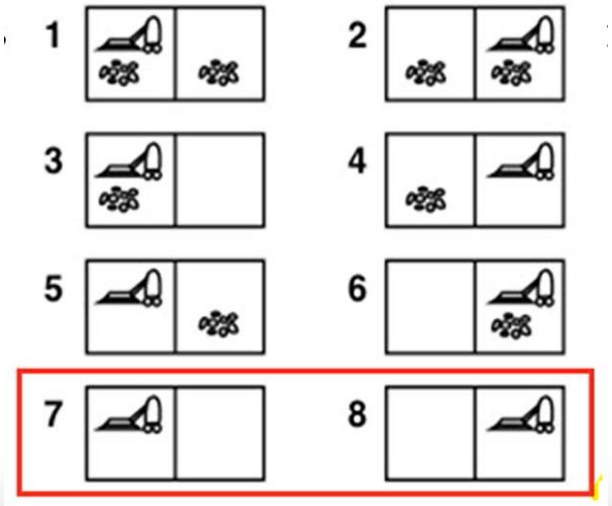
- عامل مبتنی بر حل مساله
- انواع مساله
- فرموله سازی مساله
- انواع جستجوهای ناآگاهانه

جستجوهای ناآگاهانه

عامل مبتنی بر حل مساله

- چهار مرحله کلی برای حل یک مساله عبارتند از:

1. فرموله‌سازی هدف (goal formulation): کدامیک از حالات هدف است؟



مثال: جاروبرقی هوشمند، هدف، هردو اتاق تمیز باشند و جاروبرقی در اتاق سمت راست یا چپ باشد.

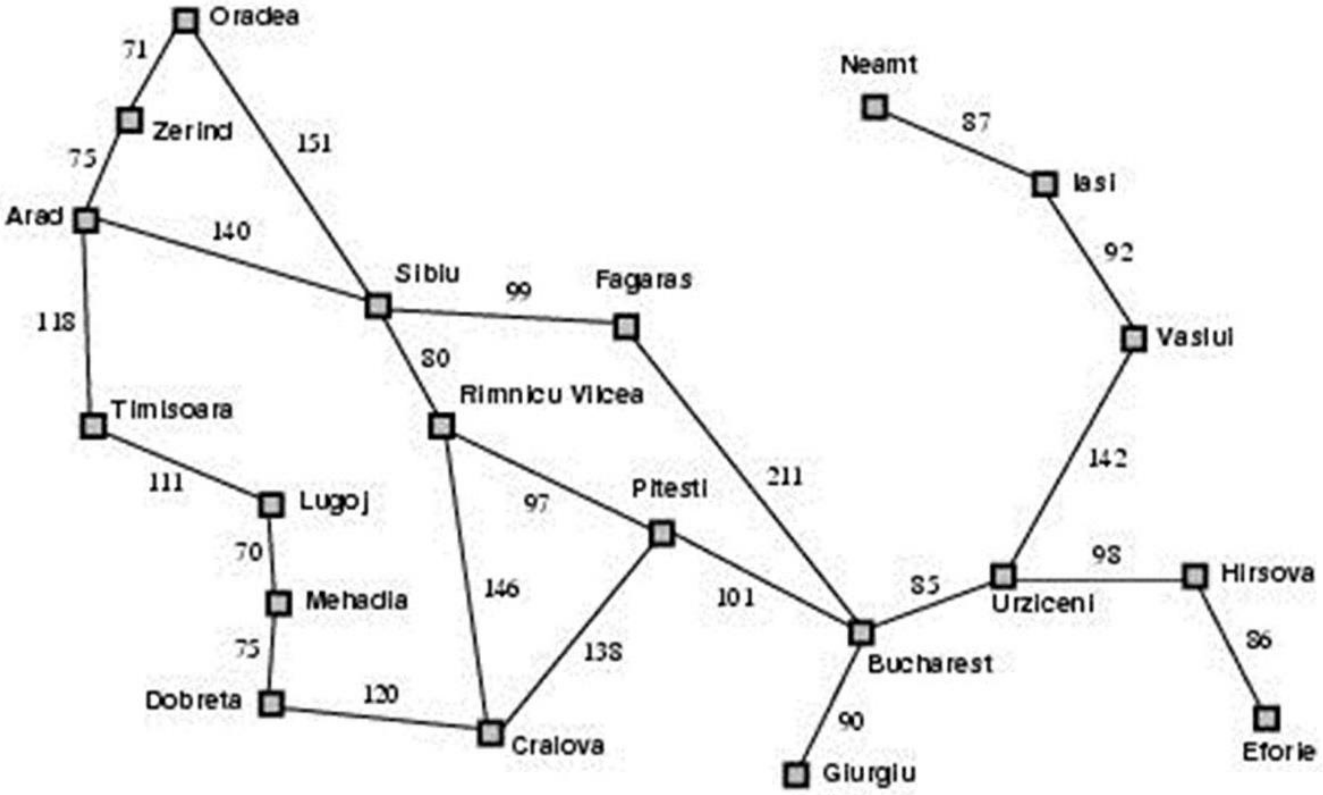
جستجوهای ناآگاهانه

2. فرموله‌سازی مساله (problem formulation): چه حالات و اعمالی برای رسیدن به هدف مورد نیاز است؟
3. جستجو (search): دنباله‌ایی از اعمال از حالت شروع تا هدف (راه حل)تعیین می‌شود.
4. اجرا (execute): با دریافت راه‌حل، اعمال مورد نظر را انجام می‌دهد.

مثال: فرض کنید درشهر آراد (Arad) در کشور رمانی هستیم و فردا بلیط برگشت از شهر بخارست (Bucharest) به کشورمان را داریم. می خواهیم بدانیم کوتاهترین مسیر از آراد به بخارست کدام است؟

جستجوهای ناآگاهانه

مثال: رمانی



جستجوهای ناآگاهانه

مثال: رمانی

- حالت شروع: شهر Arad
- فرموله سازی هدف: شهر Bucharest
- فرموله سازی مساله:
  - ✓ حالات: شهرهای مختلف
  - ✓ اعمال: حرکت بین شهرها
- جستجو: Arad, Sibiu, Fagaras, Bucharest (جواب)

جستجوهای ناآگاهانه

عامل مبتنی بر حل مساله

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) return an action
    static: seq , an action sequence
            state, some description of the current world state
            goal, a goal
            problem, a problem formulation

    state ← UPDATE-STATE(state, percept)
    if seq is empty then
            goal ← FORMULATE-GOAL(state)
            problem ← FORMULATE-PROBLEM(state , goal)
            seq ← SEARCH(problem)
    action ← FIRST(seq)
    seq ← REST(seq)
    return action
```

جستجوهای ناآگاهانه

چند سوال

1) جستجو چندبار انجام می‌شود؟ فقط یکبار
2) محیط کاملاً قابل مشاهده یا بخشی قابل مشاهده است؟ کاملا قابل مشاهده
3) محیط قطعی یا غیر قطعی است؟ محیط قطعی است، چون فقط با یکبار جستجو جواب بدست آمد.
4) محیط گسسته یا پیوسته است؟ گسسته
5) محیط اپیزودیک یا ترتیبی است؟ ترتیبی

جستجوهای ناآگاهانه

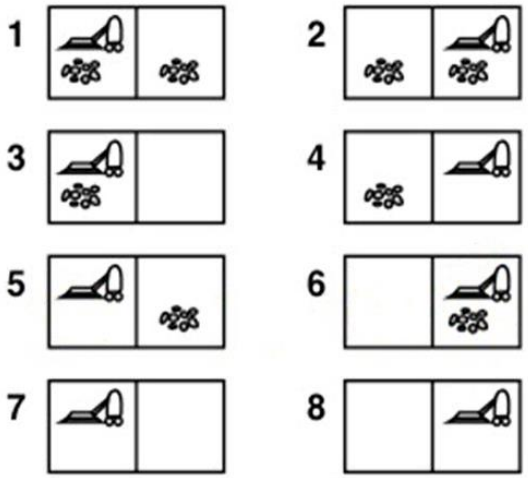
انواع مساله

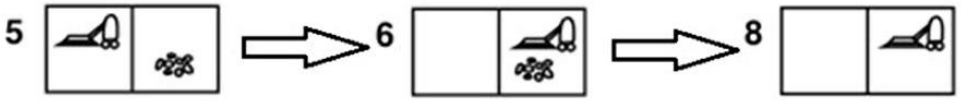
- قطعی و کاملاً قابل مشاهده: مسایل تک حالته (Single State)
- قطعی و بخشی قابل مشاهده: مسایل غیر قابل دریافت (Sensorless/Conformant)
- غیر قطعی و بخشی قابل مشاهده: مسایل احتمالی (Contingency)
- فضای حالت ناشناخته: مسایل اکتشافی یا برخط (Exploration/Online)

جستجوهای ناآگاهانه

مساله تک حالته

- مساله تک حالته : شروع از حالت ۵
- راه حل؟ [راست، مکش]



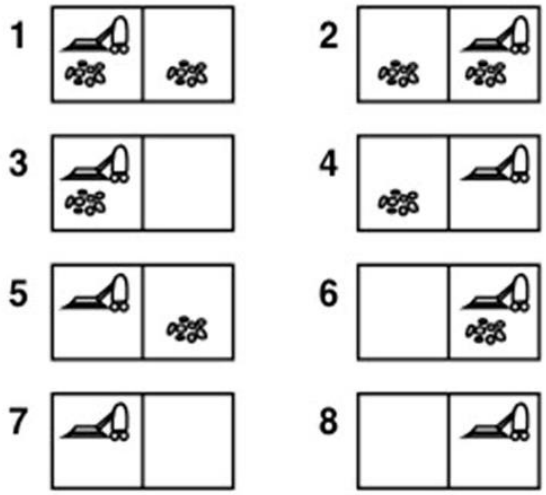


جستجوهای ناآگاهانه

مساله غیرقابل دریافت

فرض کنید یکی از دوربین ها کار نمی کند و نمی‌دانیم در کدامیک از ۸ حالت هستیم.

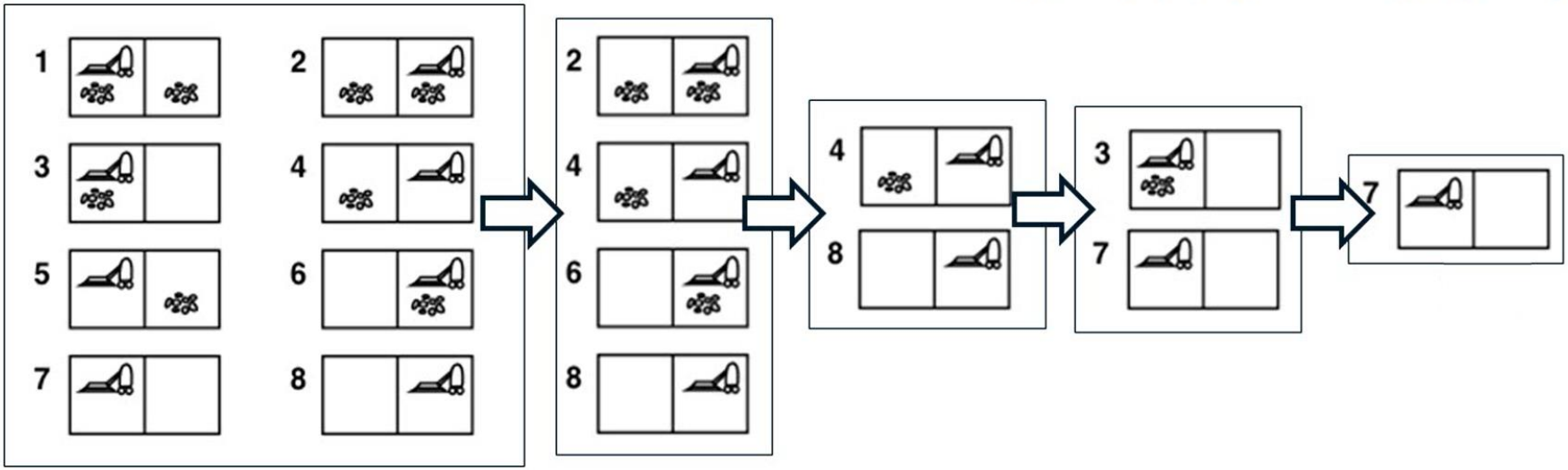
- مساله غیر قابل دریافت: شروع از یکی از حالات {۱،۲،.... ۸}
- راه حل: [راست، مکش، چپ و مکش]



جستجوهای ناآگاهانه

مساله غیرقابل دریافت

راه حل: [راست، مکش، چپ و مکش]

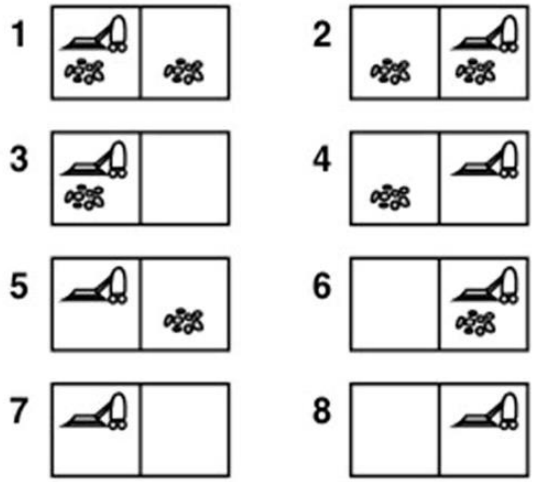


در ابتدا نمی دانیم در چه وضعیتی هستیم ولی با رعایت ترتیبی از اعمال به هدف می رسیم.

جستجوهای ناآگاهانه

مساله غیرقطعی

- مساله غیر قطعی: شروع از یکی از حالات {۳،۱}
  با فرض اینکه مکش ممکن است باعث کثیفی نیز شود



- راه حل: ؟؟

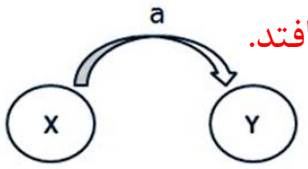
در مورد راه حل تحقیق کنید!

جستجوهای ناآگاهانه

فرموله‌سازی مساله

- یک مساله با موارد زیر تعریف می‌شود.
- حالت شروع (initial state): مانند Arad
- تابع جانشینی(successor function): مجموعه زوج مرتب از حالت – عمل
- آزمون هدف (goal test): صریح(explicit) مانند بخارست

  ضمنی(implicit) مانند مات کردن در شطرنج تحت شرایط خاصی اتفاق می‌افتد.
- هزینه مسیر(path cost): C(x, a, y) که همیشه صعودی است.

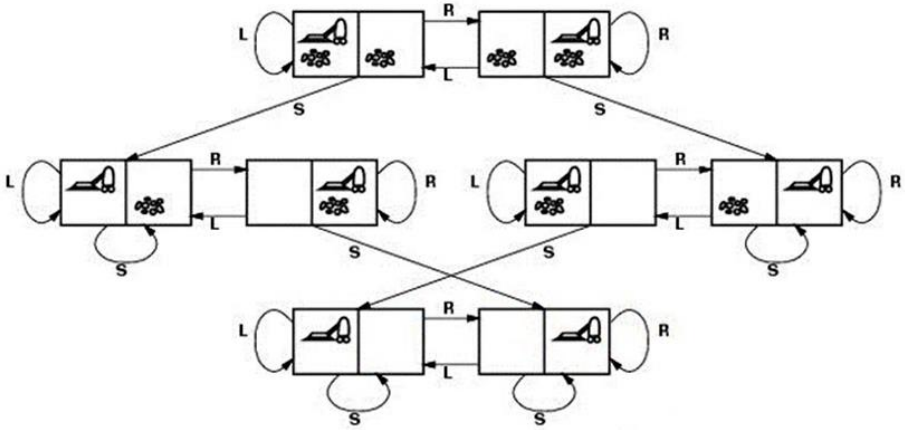


یعنی هزینه رفتن از x به y برابر a است.

- راه حل(solution): دنباله‌ایی از اعمال از حالت شروع تا حالت هدف است.
- راه حل بهینه (optimal solution): راه حل با کمترین هزینه است.

جستجوهای ناآگاهانه

مثال: دنیای جاروبرقی

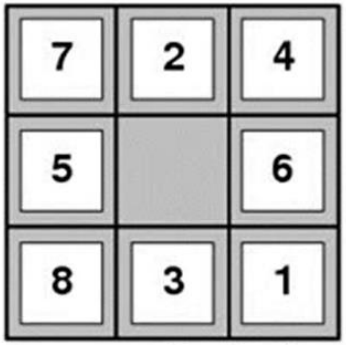


- حالات: ۸ حالت مختلف
- حالت شروع: هریک از حالات
- اعمال: {چپ، راست، مکش یا هیچ کار}
- آزمون هدف: حالات {۷ و ۸}
- هزینه مسیر: تعداد اعمال انجام شده تا رسیدن به هدف

جستجوهای ناآگاهانه

مثال: پازل اعداد

- حالات: جایگشت‌های مختلف
- حالت شروع: هریک از حالات
- اعمال: {چپ، راست، بالا و پایین}
- آزمون هدف: حالت هدف صریح بیان شده
- هزینه مسیر: تعداد اعمال انجام شده



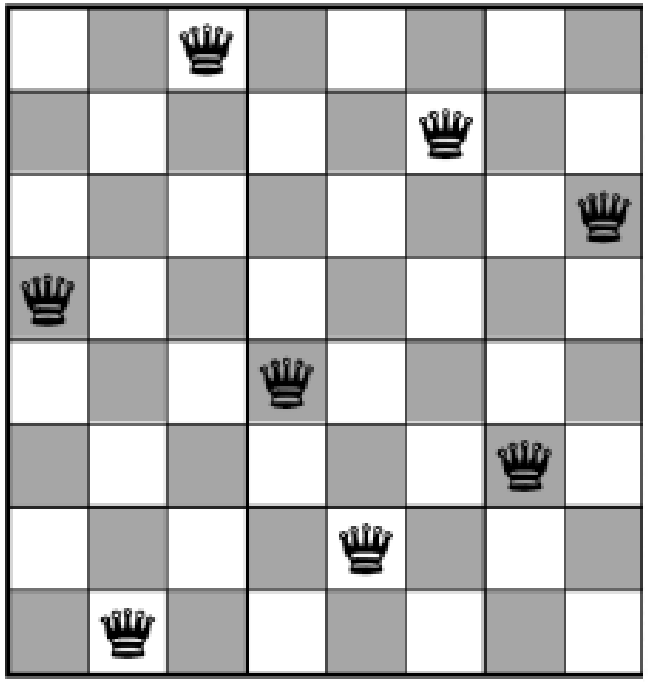
یکی از حالات شروع

هدف

جستجوهای ناآگاهانه

مثال ۸ وزیر:

هشت وزیر را چنان در صفحه شطرنج بچینیم که هیچ کدام دیگری را تهدید نکند.



حالت هدف

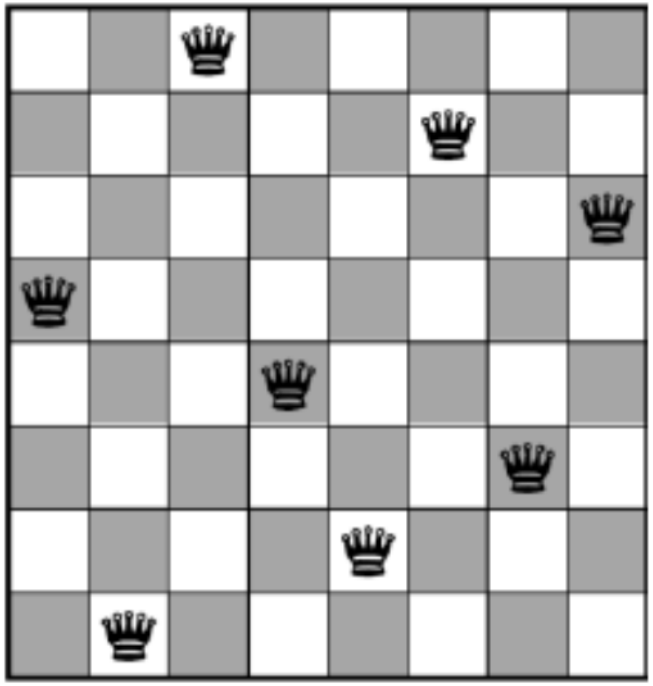
هدف ضمنی بیان شده (با یک سری شرط)

✓ ۱) فرموله سازی افزایشی

- حالات: جایگشت های مختلف چینش
- حالت شروع: صفحه خالی
- اعمال: اضافه نمودن وزیر در جای مناسب
- آزمون هدف: ۸ وزیر بر روی صفحه شطرنج
- هزینه مسیر: -

جستجوهای ناآگاهانه

مثال: ۸ وزیر



حالت هدف

✓ ۱) فرموله سازی افزایشی (روش دوم)

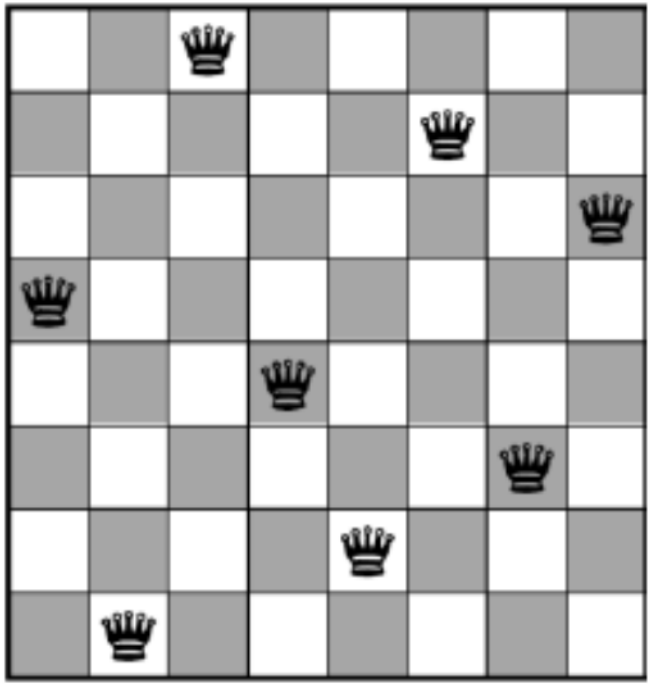
- حالات: جایگشت‌های مختلف چینش
- حالت شروع: صفحه خالی
- اعمال: اضافه نمودن هر وزیر در یک ستون
- آزمون هدف: ۸ وزیر بر روی صفحه شطرنج
- هزینه مسیر: -

❖ بهبود قابل توجه در مساله ۱۰۰ وزیر نخواهد داشت.

جستجوهای ناآگاهانه

مثال: ۸ وزیر

روش سوم

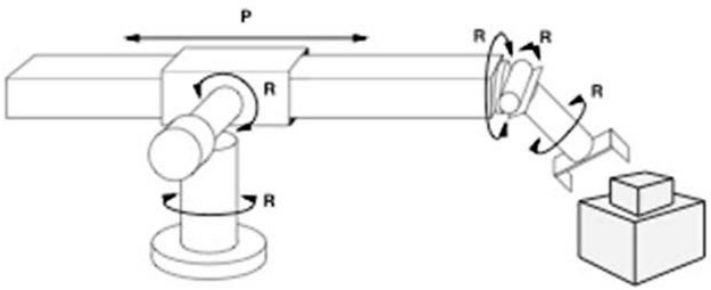


حالت هدف

- ✓ ۲) فرموله سازی کامل
- حالات: جایگشت‌های مختلف چینش
- حالت شروع: هر ۸ وزیر بر روی صفحه
- اعمال: جابجا نمودن نمودن وزیرها در صفحه
- آزمون هدف: عدم تهدید وزیرها
- هزینه مسیر: -

جستجوهای ناآگاهانه

مثال: بازوی مکانیکی قطعه ساز



- حالات: جایگشت‌های مختلف مفاصل
- حالت شروع: هر مقدار قرارگیری مفاصل
- اعمال: جابجایی مفاصل
- آزمون هدف: ساخت کامل قطعه
- هزینه مسیر: زمان اجرا

جستجوهای ناآگاهانه

الگوریتم‌های جستجوی پایه

- راه حل مسایل قبلی چگونه پیدا می‌شود؟
- جستجوی فضای حالت با تولید یک درخت
- ریشه = حالت شروع
- گره‌ها و برگ‌ها از طریق توابع جانشینی تولید می‌شوند.
- در حالت کلی جستجو منجر به تولید گراف می‌گردد.

(حالات یکسان از طریق چندین مسیر)

جستجوهای ناآگاهانه

عامل مبتنی بر حل مساله

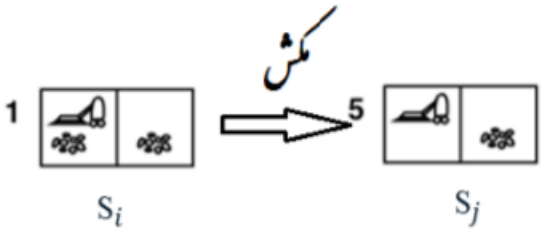
```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) return an action
    static: seq , an action sequence
            state, some description of the current world state
            goal, a goal
            problem, a problem formulation

    state ← UPDATE-STATE(state, percept)
    if seq is empty then
            goal ← FORMULATE-GOAL(state)
            problem ← FORMULATE-PROBLEM(state , goal)
            seq ← SEARCH(problem)
    action ← FIRST(seq)
    seq ← REST(seq)
    return action
```

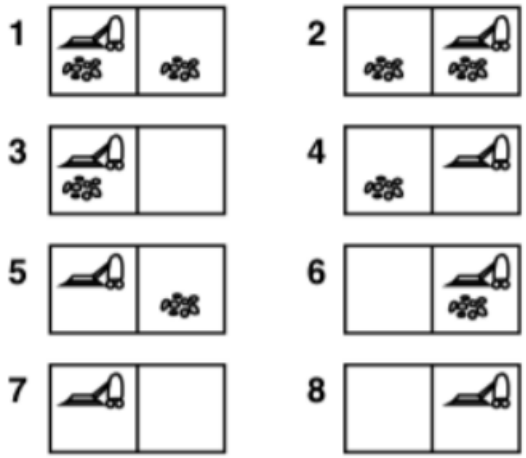
تمرکز ما در این مبحث

جستجوهای ناآگاهانه

یادآوری: عامل و محیط

- بنابراین هر محیط دارای مجموعه ای از حالت‌ها می‌باشد.
- محیط در هر لحظه در یکی از این حالت‌ها می‌باشد.
- عمل عامل در محیط باعث تغییر حالت محیط می‌شود.
- حالت فعلی: S_i
- عمل: Suck
- حالت بعدی : S_j

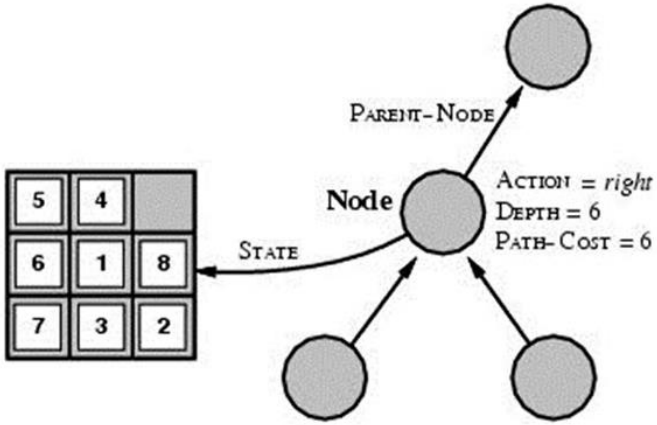




جستجوهای ناآگاهانه

فضای حالت

هر حالت (state) در حقیقت بیانگر یک حالت فیزیکی است.



برای مثال در پازل اعداد هر عدد به سمت چپ ، راست یا بالا وپایین منجر به یک حالت فیزیکی می‌شود.

جستجوهای ناآگاهانه

استراتژی‌های جستجو

- استراتژی ⟸ ترتیب بسط گره‌ها را مشخص می‌کند.
- کارایی الگوریتم جستجو (استراتژی) با چهار معیار اندازه‌گیری می‌شود.

1) کامل بودن: آیا الگوریتم همیشه راه حل را در صورت وجود آن پیدا می‌کند؟
2) بهینه بودن: آیا الگوریتم همیشه راه حل با کمترین هزینه را در صورت وجود آن پیدا می کند؟
3) پیچیدگی زمانی: چه تعداد گره در حین جستجو تولید/ بسط داده می‌شود؟
4) پیچیدگی حافظه: چه تعداد گره در حین جستجو در حافظه ذخیره می‌شود؟

جستجوهای ناآگاهانه

استراتژی‌های جستجو

- پیچیدگی زمانی و حافظه معمولاً با سه پارامتر تعریف می‌شود:
- (b): حداکثر فاکتور انشعاب
- (d): عمق جواب بهینه
- (m): حداکثر عمق فضای حالت (معمولاً بینهایت است)

جستجوهای ناآگاهانه

جستجوهای ناآگاهانه

- جستجوی ناآگاهانه فقط از اطلاعات موجود در صورت مساله استفاده می‌نماید (جستجوهای کورکورانه).

- اگر استراتژی جستجو بتواند حالات غیر هدف را از حالات هدف تشخیص دهد، آنگاه جستجو آگاهانه نامیده می شود (فصل چهارم).

  - الگوریتم ناآگاهانه:الگوریتم‌های این فصل همگی بجزء تعریف مساله،دانش دیگری در رابطه با مساله در اختیار ندارند.به همین خاطر به آنها الگوریتم ناآگاهانه گویند.

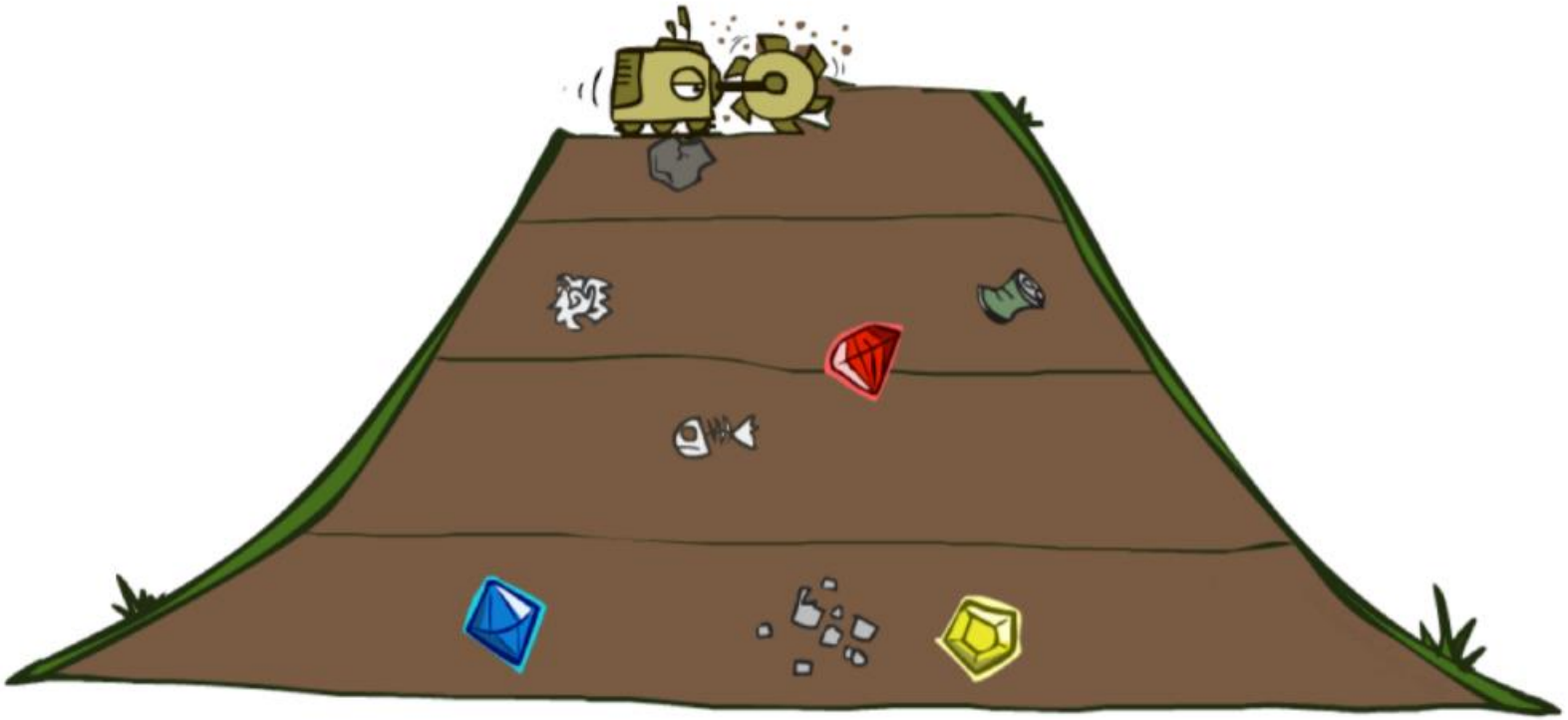
جستجوهای ناآگاهانه

انواع جستجوهای ناآگاهانه

1. جستجوی سطحی (Breadth-first search)
2. جستجوی هزینه یکنواخت (Uniform-cost search)
3. جستجوی عمقی (Depth-first search)
4. جستجوی عمقی محدود (Depth-limited search)
5. جستجوی عمقی تکرارشونده (Iterative deepening search)
6. جستجوی دوطرفه (Bidirectional search)

جستجوهای ناآگاهانه

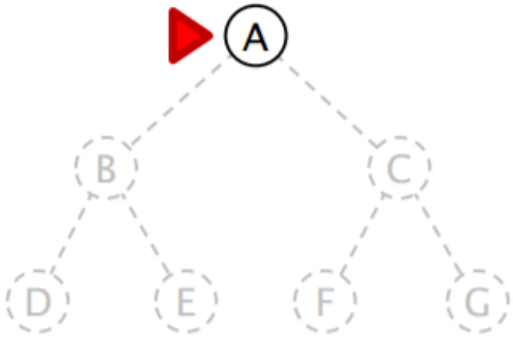
جستجوی سطحی



جستجوهای ناآگاهانه

جستجوی سطحی

- جستجوی سطحی.
  - هر بار سطحی‌ترین گره گسترش نیافته را گسترش می‌دهد.



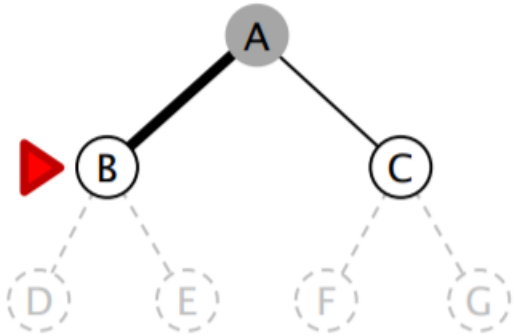
fringe: [A]

- پیاده‌سازی.
  - fringe یک صف FIFO است.
  - فرزندان جدید به انتهای صف اضافه می‌شوند.

جستجوهای ناآگاهانه

جستجوی سطحی

- جستجوی سطحی.
  - هر بار سطحی‌ترین گره گسترش نیافته را گسترش می‌دهد.



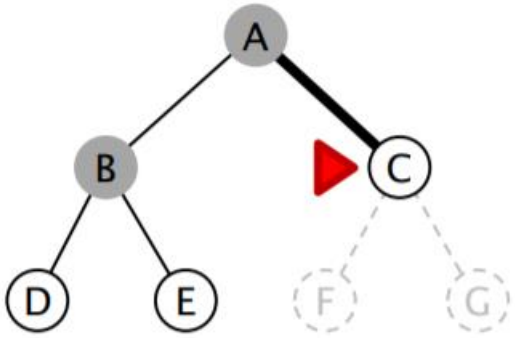
fringe: [B, C]

- پیاده‌سازی.
  - fringe یک صف FIFO است.
  - فرزندان جدید به انتهای صف اضافه می‌شوند.

جستجوهای ناآگاهانه

جستجوی سطحی

- جستجوی سطحی.
  - هر بار سطحی‌ترین گره گسترش نیافته را گسترش می‌دهد.



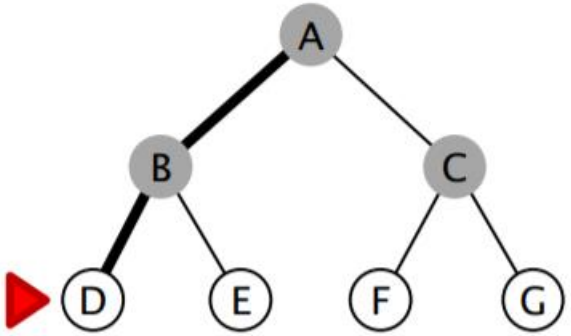
fringe: [C, D, E]

- پیاده‌سازی.
  - fringe یک صف FIFO است.
  - فرزندان جدید به انتهای صف اضافه می‌شوند.

جستجوهای ناآگاهانه

جستجوی سطحی

- جستجوی سطحی.
  - هر بار سطحی‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی.
  - fringe یک صف FIFO است.
  - فرزندان جدید به انتهای صف اضافه می‌شوند.



fringe: [D, E, F, G]

جستجوهای ناآگاهانه

جستجوی سطحی

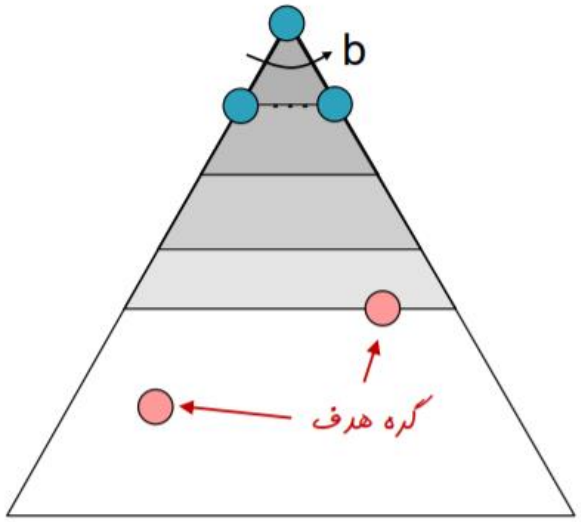
```
function BREADTH-FIRST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  frontier ← a FIFO queue with node as the only element
  explored ← an empty set
  loop do
    if 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 not in explored or frontier then
        if problem.GOAL-TEST(child.STATE) then return SOLUTION(child)
        frontier ←INSERT(child, frontier)
```

- توجه. آزمایش هدف در لحظه‌ی تولید گره‌ها انجام می‌شود و نه در لحظه‌ی گسترش.

جستجوهای ناآگاهانه

ارزیابی جستجوی سطحی

- جستجوی سطحی کدام گره‌ها را گسترش می‌دهد؟
- تمام گره‌های بالاتر از کم عمق‌ترین گره راه‌حل!



1

b

b^2

b^d

جستجوهای ناآگاهانه

ارزیابی جستجوی سطحی

- کامل؟ **بله** [اگر فاکتور انشعاب محدود باشد]
- بهینه؟ **خیر** [مگر این که تابع هزینه‌ی مسیر برحسب عمق گره‌ها غیرنزولی باشد]
- پیچیدگی زمانی؟ نمایی

$$b^1 + b^2 + b^3 + \cdots + b^d \in O(b^d)$$

- پیچیدگی حافظه؟ نمایی

$$O(b^{d-1}) + O(b^d) \in O(b^d)$$

تعداد گره های *explored*

تعداد گره های *frontier*

جستجوهای ناآگاهانه

ارزیابی جستجوی سطحی

- مصرف زمان و حافظه در جستجوی سطحی.

عمق	تعداد گره ها	زمان	حافظه
۲	۱۱۰	۰/۱۱ میلی ثانیه	۱۰۷ کیلو بایت
۴	۱۱۱۱۰	۱۱ میلی ثانیه	۱۰/۶ مگابایت
۶	$۱۰^{۶}$	۱/۱ ثانیه	۱ گیگابایت
۸	$۱۰^{۸}$	۲ دقیقه	۱۰۳ گیگابایت
۱۰	$۱۰^{۱۰}$	۳ ساعت	۱۰ ترابایت
۱۲	$۱۰^{۱۲}$	۱۳ روز	۱ پتابایت
۱۴	$۱۰^{۱۴}$	۳/۵ سال	۹۹ پتابایت
۱۶	$۱۰^{۱۶}$	۳۵۰ سال	۱۰ هگزابایت

فاکتور انشعاب = ۱۰؛ یک میلیون گره بر ثانیه؛ ۱۰۰۰ بایت به ازای هر گره

جستجوهای ناآگاهانه

ارزیابی جستجوی سطحی

سوال: آیا هر جستجوی کاملی بهینه است؟
جواب: خیر

سوال: برعکس، آیا هر جستجوی بهینه کامل است؟
جواب: بلی

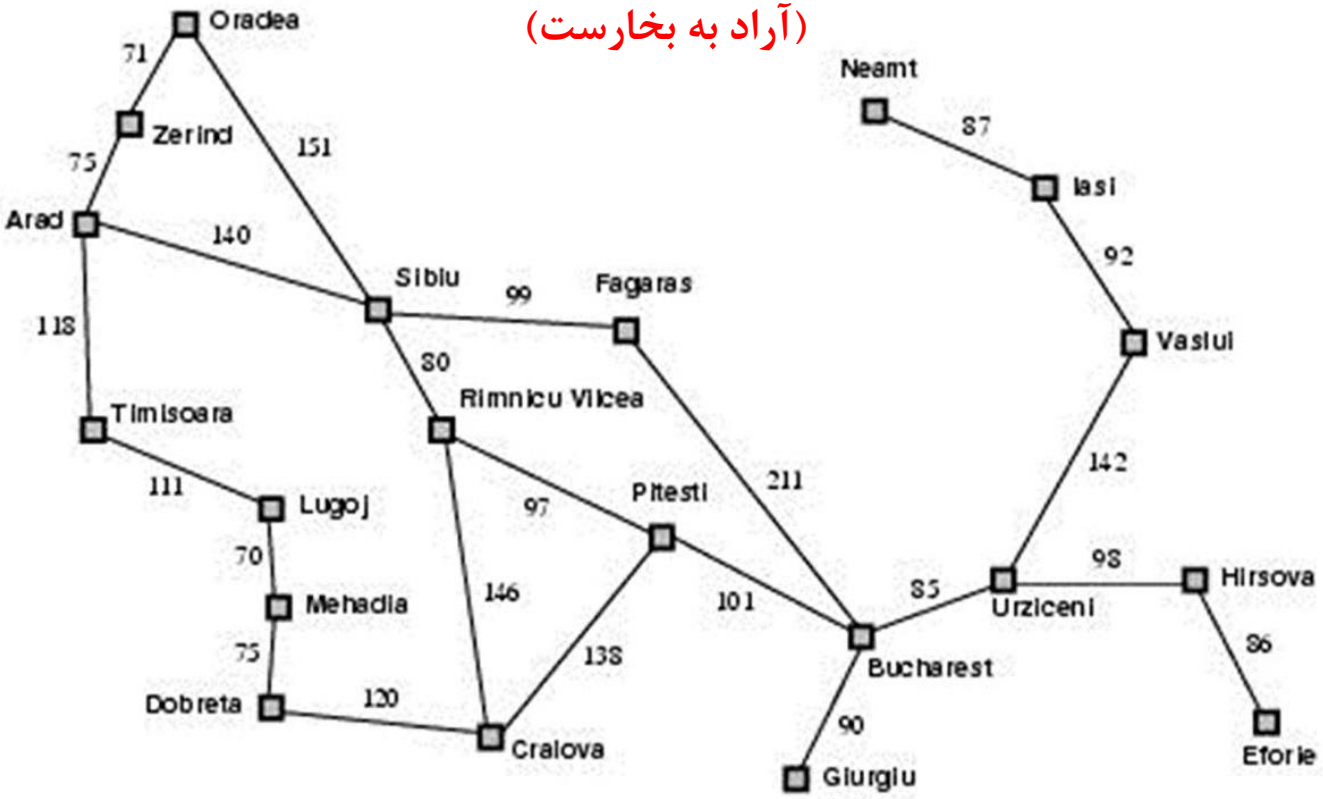
سوال: اگر جستجوکامل نباشد بهینه هم نیست؟
جواب: بلی

سوال: اگر جستجو بهینه نباشد کامل هم نیست؟
جواب: خیر، ممکن است کامل باشد ولی بهینه نباشد.

جستجوهای ناآگاهانه

مثال: رمانی

(آراد به بخارست)



جستجوهای ناآگاهانه

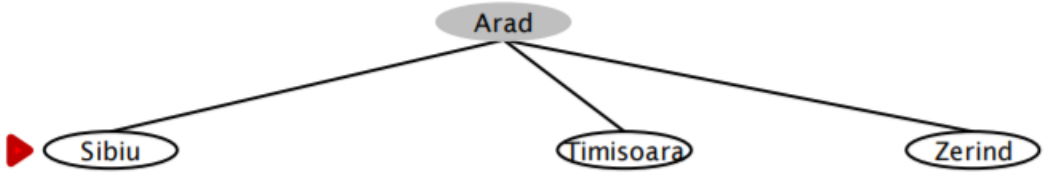
مثال: (آراد به بخارست)

frontier: [Arad]

explored: {}

جستجوهای ناآگاهانه

مثال: (آراد به بخارست)

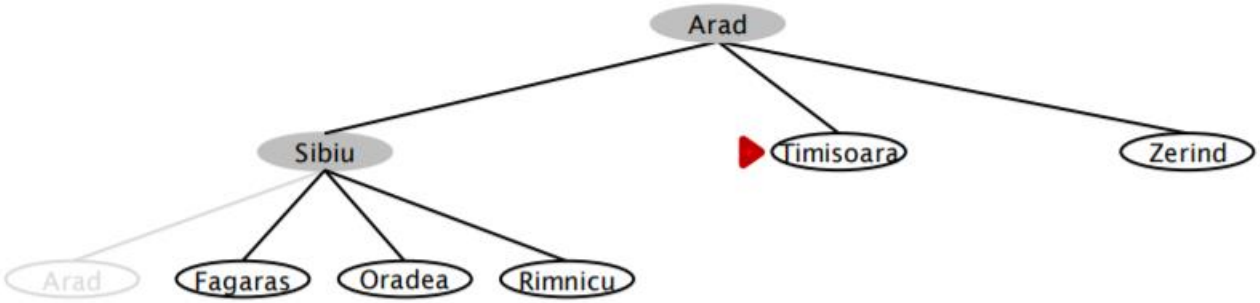


frontier: [Sibiu, Timisoara, Zerind]

explored: {Arad}

جستجوهای ناآگاهانه

مثال: (آراد به بخارست)

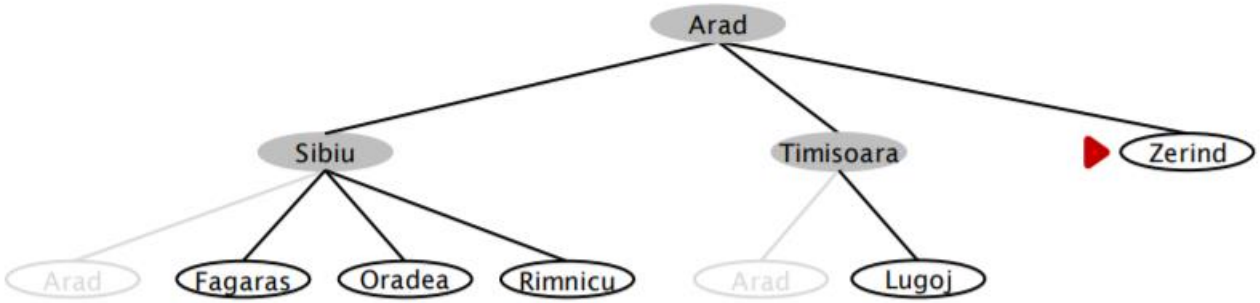


frontier: [Timisoara, Zerind, Fagaras, Oradea, Rimnicu]

explored: {Arad, Sibiu}

جستجوهای ناآگاهانه

مثال: (آراد به بخارست)

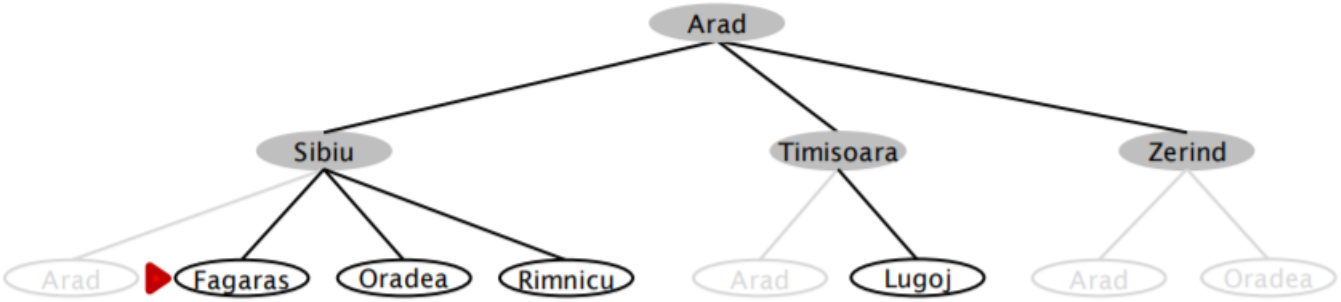


frontier: [Zerind, Fagaras, Oradea, Rimnicu, Lugoj]

explored: {Arad, Sibiu, Timisoara}

جستجوهای ناآگاهانه

مثال: (آراد به بخارست)

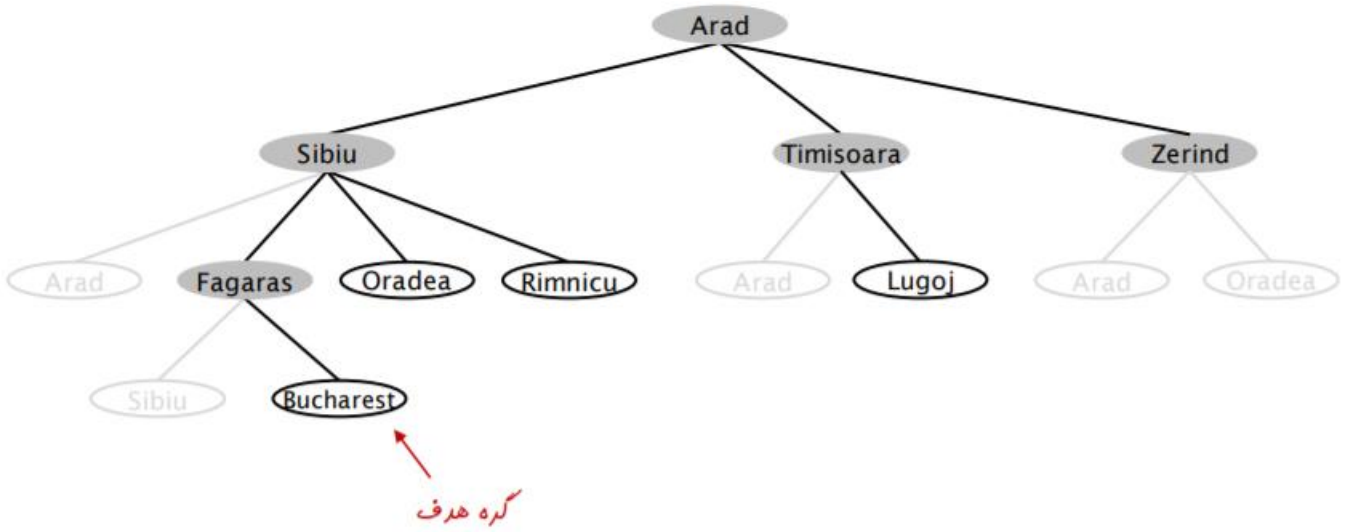


frontier: [Fagaras, Oradea, Rimnicu, Lugoj]

explored: {Arad, Sibiu, Timisoara, Zerind}

جستجوهای ناآگاهانه

مثال: (آراد به بخارست)

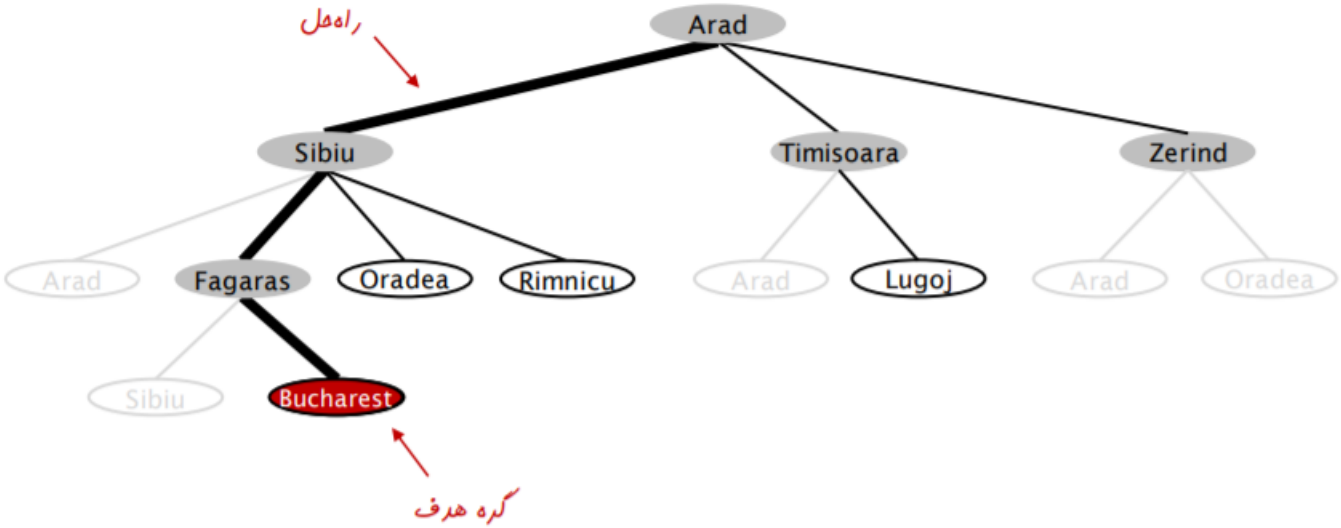


frontier: [Oradea, Rimnicu, Lugoj]

explored: {Arad, Sibiu, Timisoara, Zerind, Fagaras}

جستجوهای ناآگاهانه

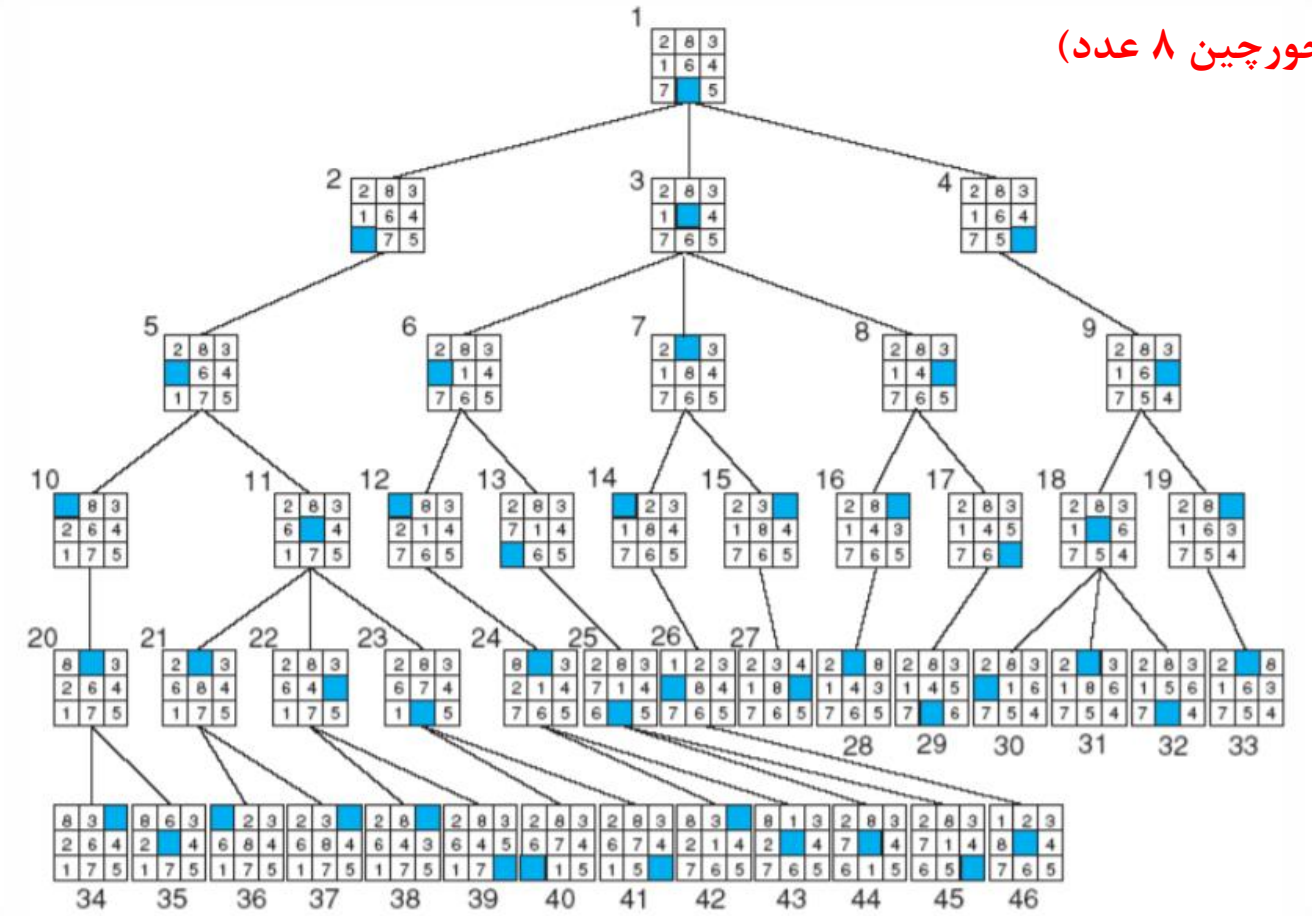
مثال: (آراد به بخارست)



جواب: Arad → Sibiu → Fagaras → Bucharest

جستجوهای ناآگاهانه

مثال: جستجوی سطحی (جورچین ۸ عدد)



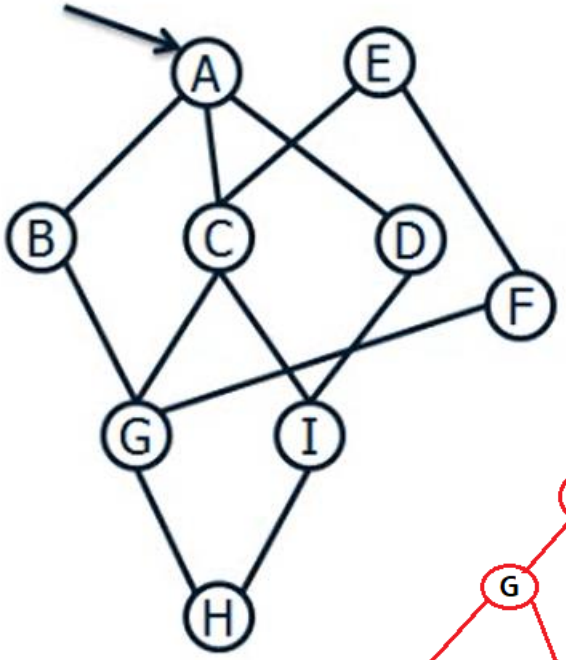
1	2	3
4	5	6
7	8	

هدف

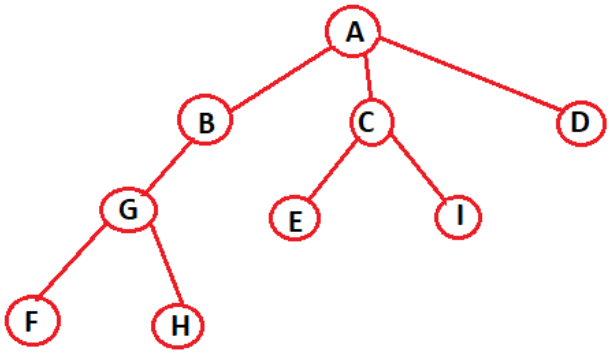
جستجوهای ناآگاهانه

مثال

درصورتیکه فضای جستجودریک مساله بصورت گراف زیر باشد نتیجه جستجوی BFSرا بیابید.



درخت جستجو



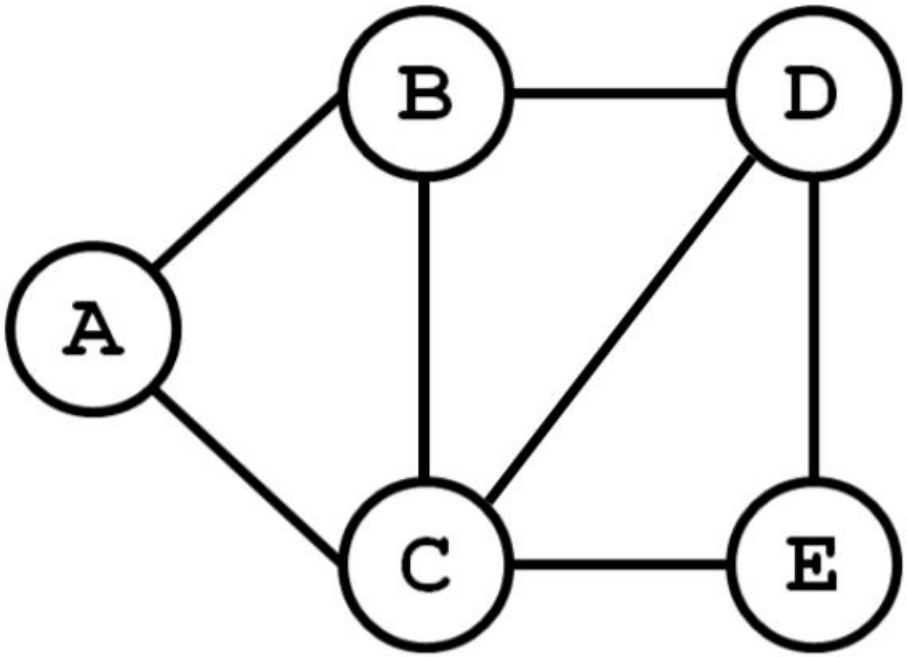
صف	جستجو
BCD	A
CDG	B
DGEI	C
GEI	D
EIFH	G
IFH	E
FH	I
H	F
	H

جستجوهای ناآگاهانه

تمرین:

کدام مورد نمی‌تواند ترتیب اجرای الگوریتم BFS روی گراف زیر باشد؟

1. CEDAB
2. BCDAE
3. ABCED
4. ACBDE



جستجوهای ناآگاهانه

جستجوی سطری

- مثال) در یک مساله با فاکتور انشعاب ۲ و عمق جواب بهینه ۴، جستجوی سطحی دقیقاً چه تعداد گره تولید می‌شود؟

$$2^0 + 2^1 + 2^2 + 2^3 + 2^4 + (2^{4+1} - 2) = 61$$

- چه تعداد گره بسط داده می‌شود؟

$$2^{4+1} - 2$$

جستجوهای ناآگاهانه

تمرین:

فرض کنید برای مسئله‌ای با جستجوی اول پهنا (breadth first) و تست هدف در لحظه تولید نیاز به بسط دادن (expand) ۳۲ گره باشد. اگر فاکتور انشعاب (branching factor) درخت جستجو ثابت باشد و عمق درخت برابر ۵ و عمق هدف برابر ۴ باشد، کدامیک از گزینه‌ها مقدار فاکتور انشعاب (b) را نشان می‌دهد؟ (فرض کنید ریشه درخت در عمق صفر (۰) واقع شده است)

1. $b = 2$
2. $b > 5$
3. $2 < b < 3$
4. $3 \leq b \leq 5$

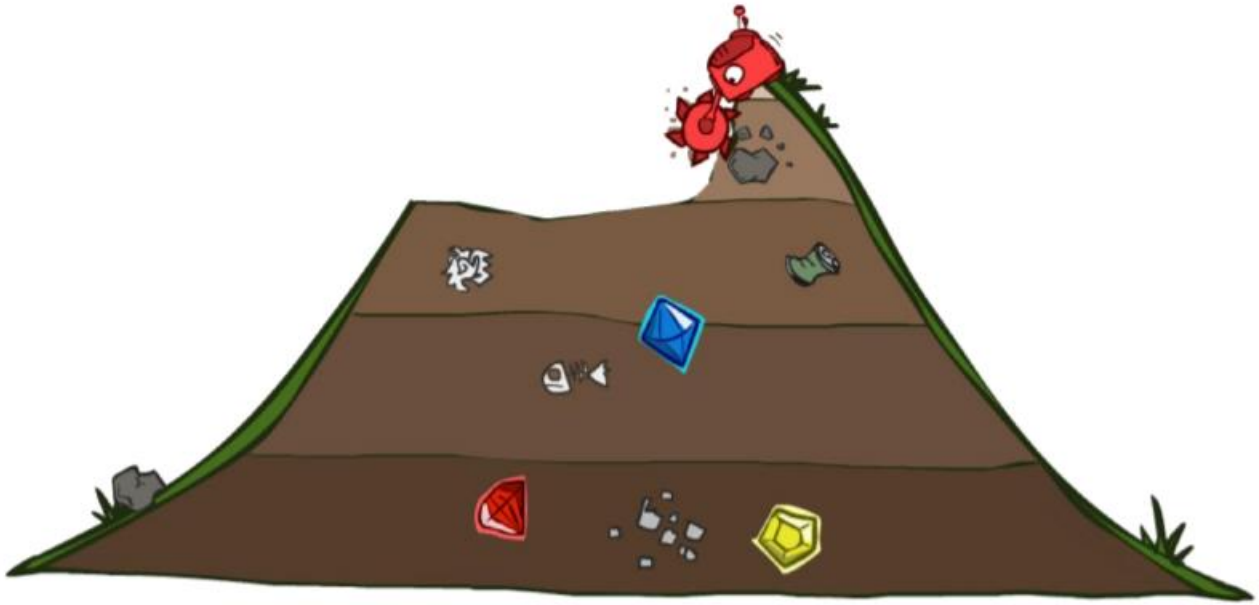
جستجوهای ناآگاهانه

تمرین:

ضریب انشعاب یک درخت جستجو ۳ می باشد. حل مسأله در آخرین رأسی که در عمق ۲ جستجو می شود وجود دارد. چه تعداد رأس باید بسط داده شوند تا این رأس بازدید شود در صورتی که از جستجوی عرض نخست استفاده شود؟ (فرض بر این است که حل مسأله بودن یک گره، در زمان باز کردن فرزندان آن گره بررسی می‌گردد.)

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت



جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- جستجوی هزینه یکنواخت.
  - هر بار کم هزینه‌ترین گره گسترش نیافته را گسترش می‌دهد.
  - هزینه‌ی گره: هزینه‌ی مسیر از ریشه تا آن گره در درخت جستجو.

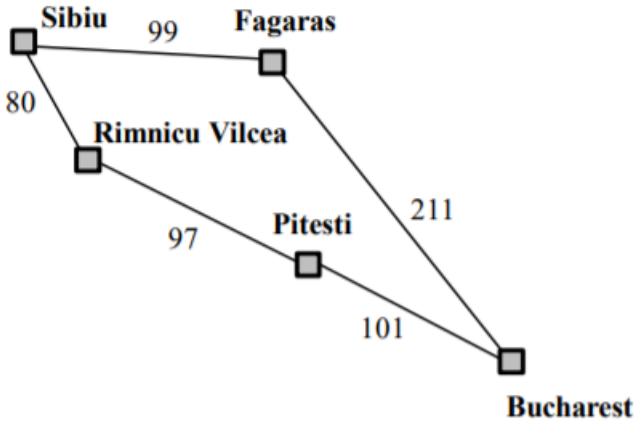
- پیاده سازی.
  - fringe یک صف اولویت است.
  - اولویت یک گره: هزینه‌ی مسیر از ریشه تا آن گره.

به جستجو با هزینه یکنواخت بطور اختصار UCS نیز گفته می‌شود.

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.

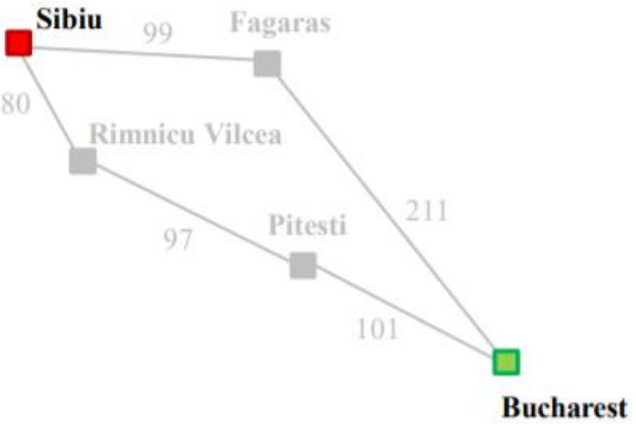


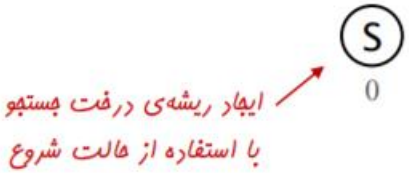

```
frontier: []
explored: {}
```

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.



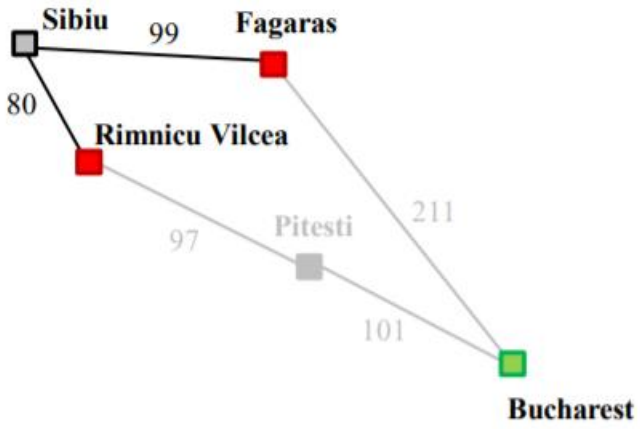


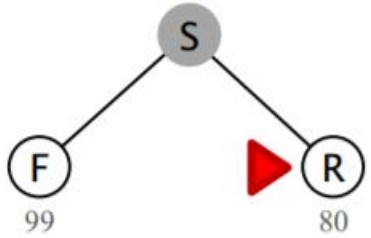

```
frontier: [Sibiu (0)]
explored: {}
```

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.





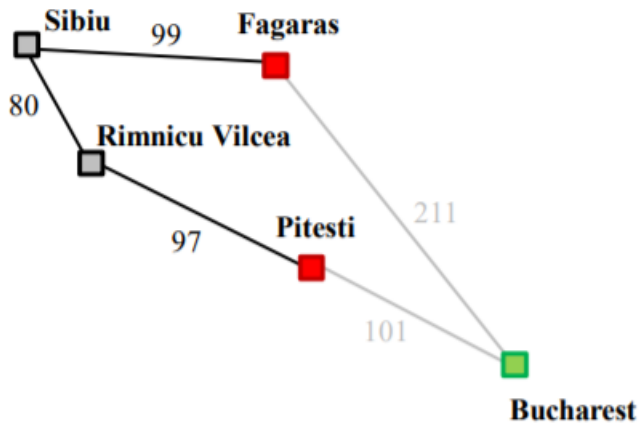
گسترش گره ریشه و افزودن
فرزندان آن به صف اولویت

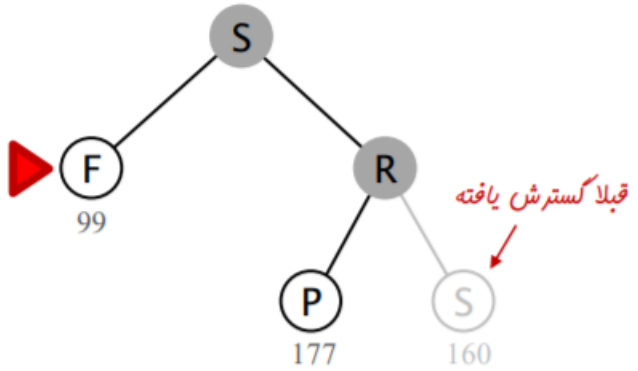
```
frontier: [Rimnicu (80), Fagaras (99)]
explored: {Sibiu}
```

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.





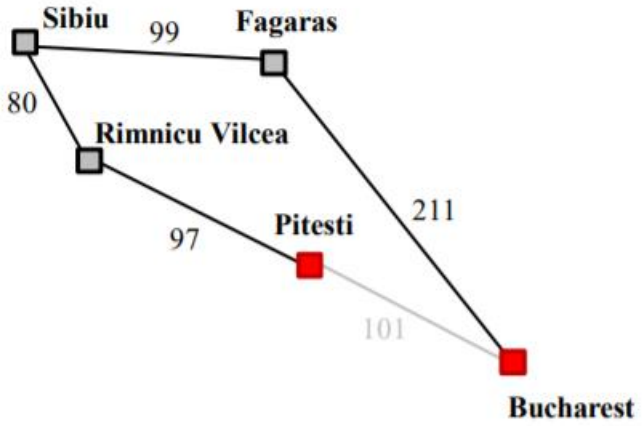
گسترش گره R و افزودن
فرزندان آن به صف اولویت

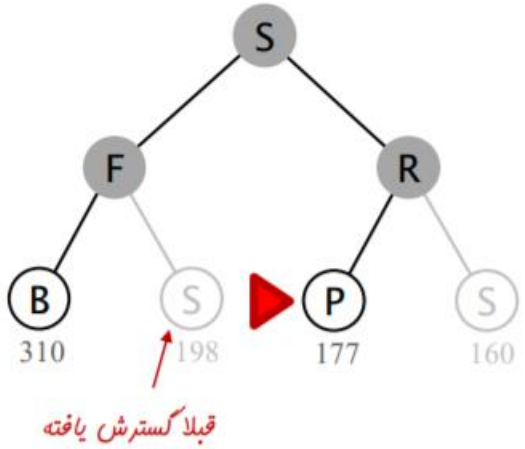
```
frontier: [Fagaras (99), Pitesti (177)]
explored: {Sibiu, Rimnicu}
```

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.





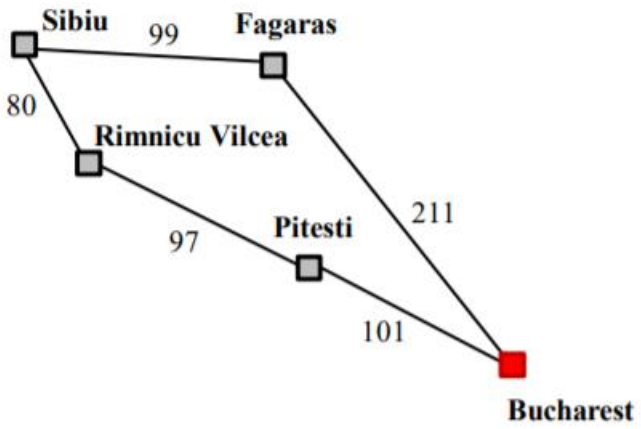
frontier: [Pitesti (177), Bucharest (310)]

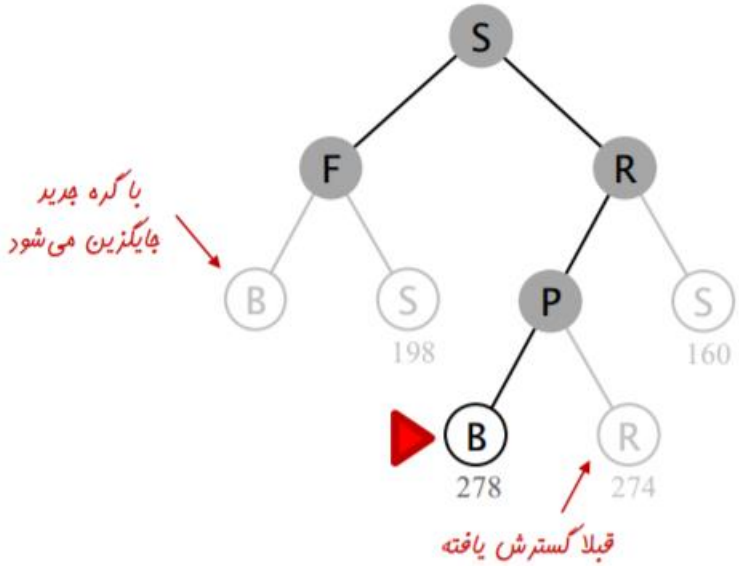
explored: {Sibiu, Rimnicu, Fagaras}

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.



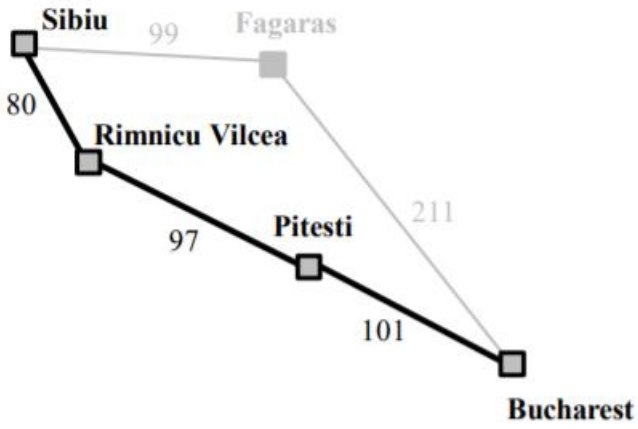


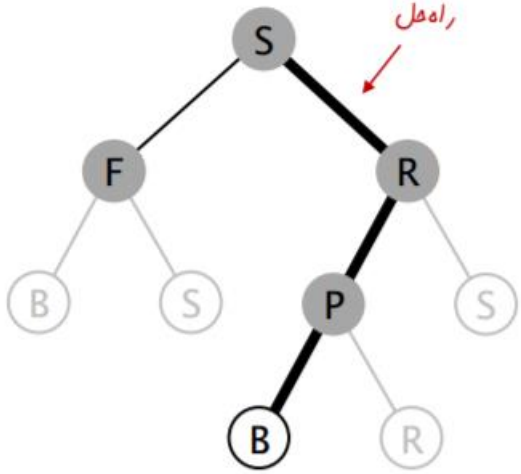

```
frontier: [Bucharest (278)]
explored: {Sibiu, Rimnicu, Fagaras, Pitesti}
```

جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

- مثال. یافتن مسیر از سیبیو به بخارست.





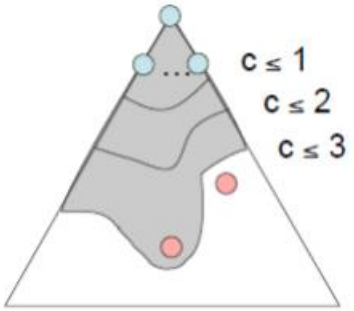
جستجوهای ناآگاهانه

جستجوی با هزینه یکنواخت

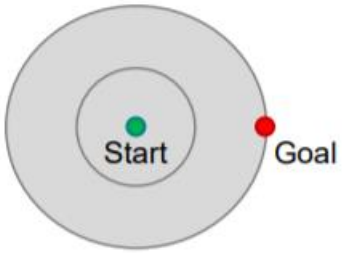
```
function UNIFORM-COST-SEARCH(problem) returns a solution, or failure
  node ← a node with STATE = problem.INITIAL-STATE, PATH-COST = 0
  frontier ← a priority queue ordered by PATH-COST, with node as the only element
  explored ← an empty set
  loop do
    if EMPTY?(frontier) then return failure
    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 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
```

جستجوهای ناآگاهانه

ارزیابی جستجوی با هزینه یکنواخت



- کامل؟ بله [اگر هزینه‌ی تمام عمل‌ها مثبت باشد]
- بهینه؟ بله [اگر هزینه‌ی تمام عمل‌ها مثبت باشد]
- پیچیدگی زمانی؟ نمایی $O(b^{1+\lfloor C^*/\epsilon \rfloor})$
- پیچیدگی حافظه؟ نمایی $O(b^{1+\lfloor C^*/\epsilon \rfloor})$



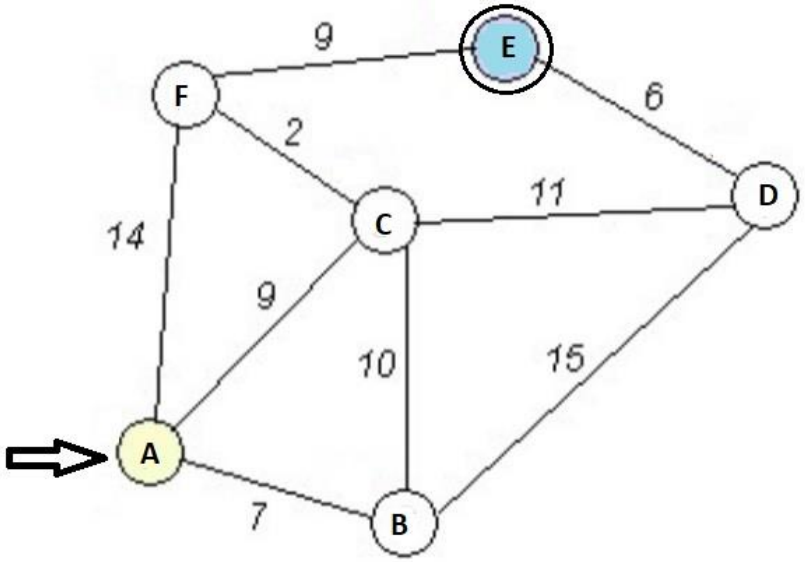
فرض کنید C^* هزینه جواب بهینه باشد و فرض کنید هزینه هر فعالیت حداقل ε است

جستجوی هزینه یکسان بدون توجه به هدف، در همه‌ی جهت‌ها جستجو می‌کند.

جستجوهای ناآگاهانه

تمرین:

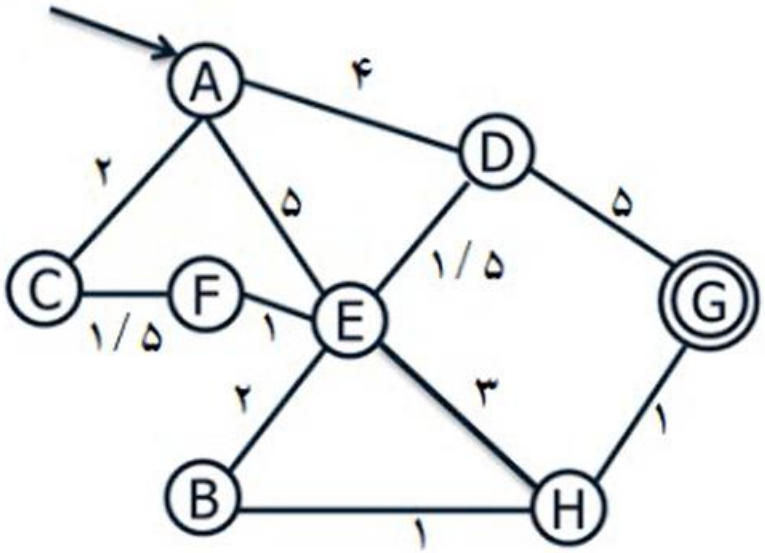
درصورتیکه فضای جستجودریک مساله بصورت گراف زیر باشد نتیجه جستجوی UCSرا بیابید.



جستجوهای ناآگاهانه

تمرین:

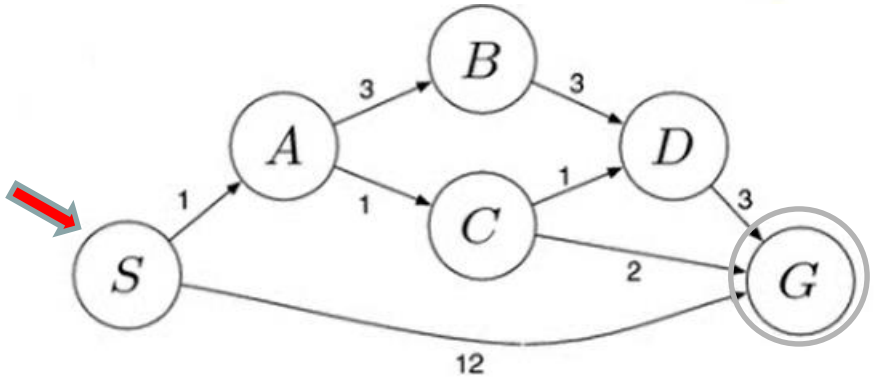
درصورتیکه فضای جستجودریک مساله بصورت گراف زیر باشد نتیجه جستجوی UCSرا بیابید.



جستجوهای ناآگاهانه

تمرین:

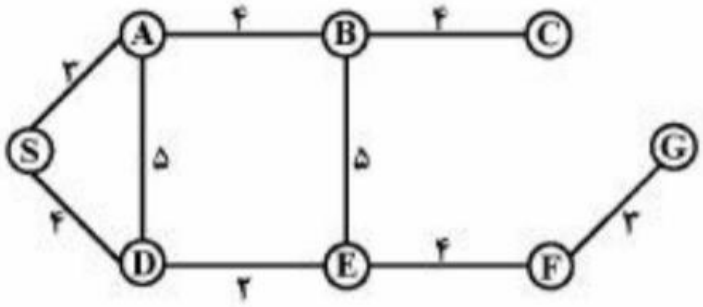
درصورتیکه فضای جستجودریک مساله بصورت گراف زیر باشد نتیجه جستجوی UCSرا بیابید.



جستجوهای ناآگاهانه

تمرین:

در مسألهٔ زیر برای رسیدن از S به G با استفاده از روش جستجوی هزینهٔ یکنواخت (uniform cost search) در حالت جستجوی گرافی، کدام گره‌ها به‌ترتیب پیمایش می‌شوند؟



۱) SDEFG

۲) SADEFG

۳) SADEBFG

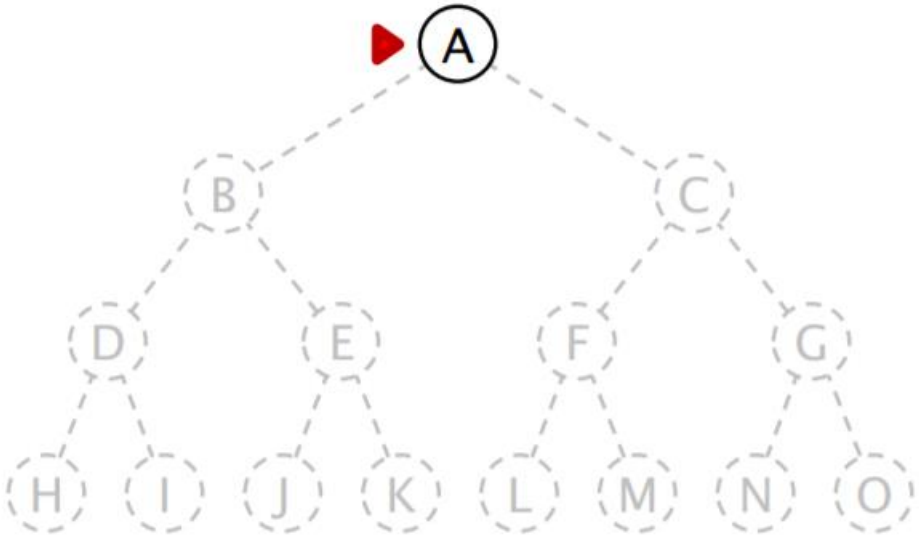
۴) SADEBFCG

جستجوهای ناآگاهانه

جستجوی عمقی

جستجوهای ناآگاهانه

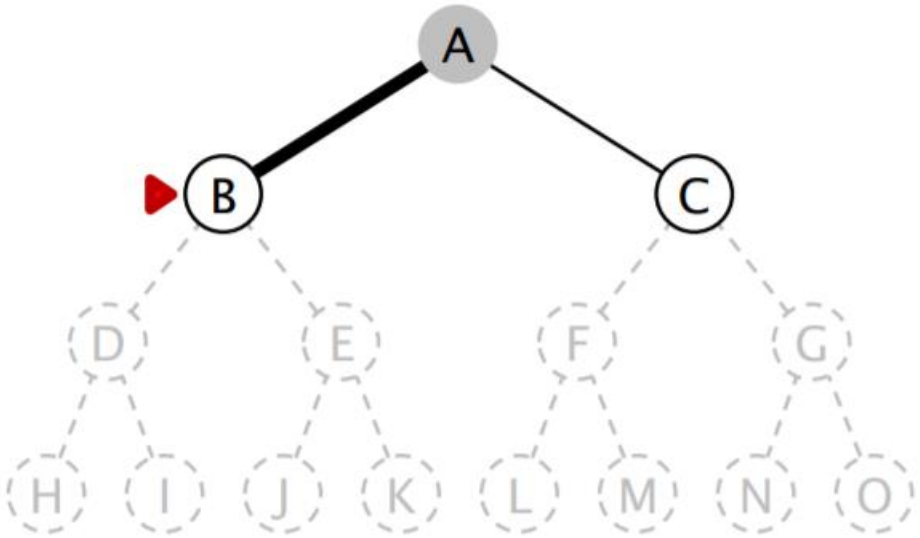
- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



Fringe=[A]

جستجوهای ناآگاهانه

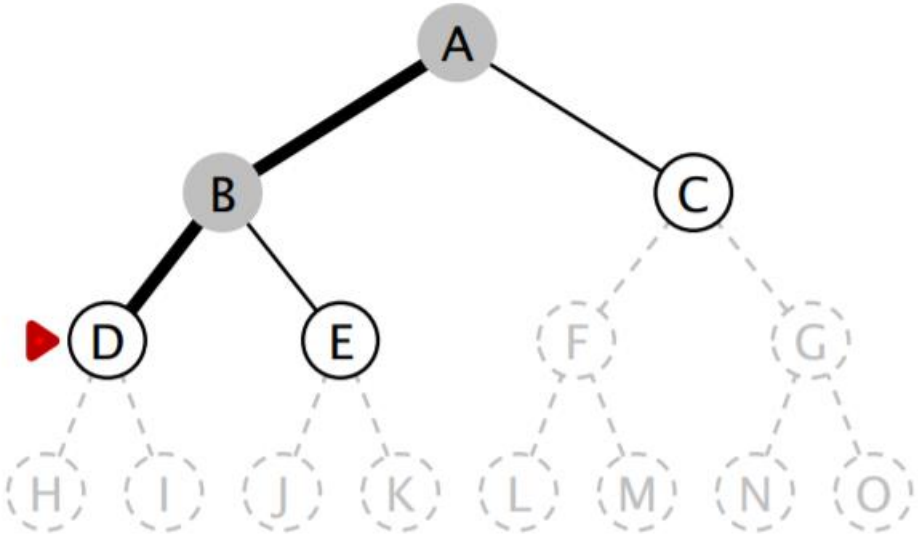
- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



Fringe=[B,C]

جستجوهای ناآگاهانه

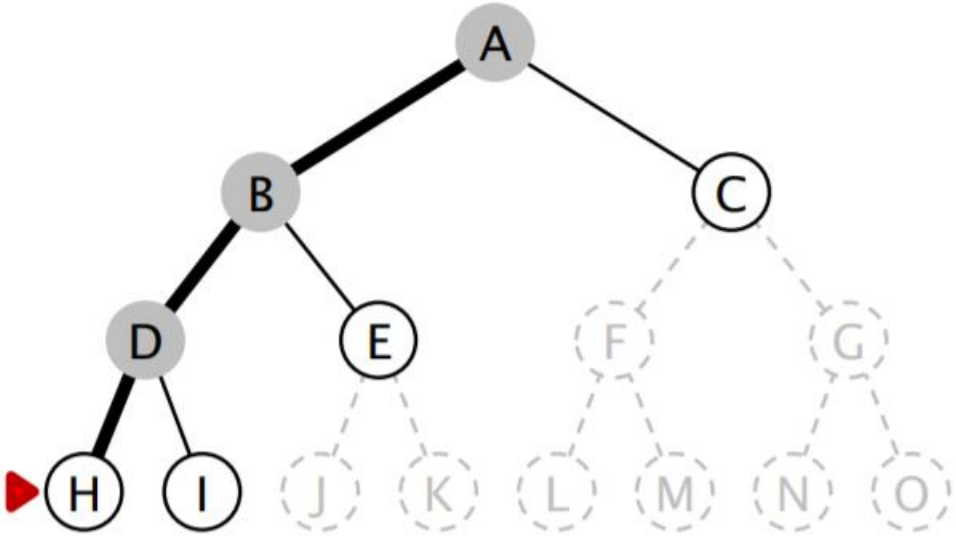
- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



Fringe=[D,E,C]

جستجوهای ناآگاهانه

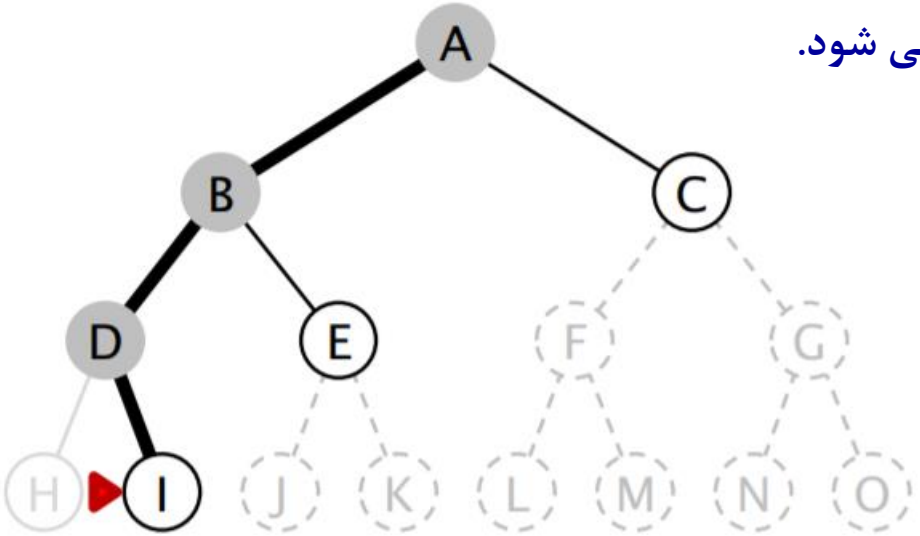
- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



Fringe=[H,I,E,C]

جستجوهای ناآگاهانه

- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.

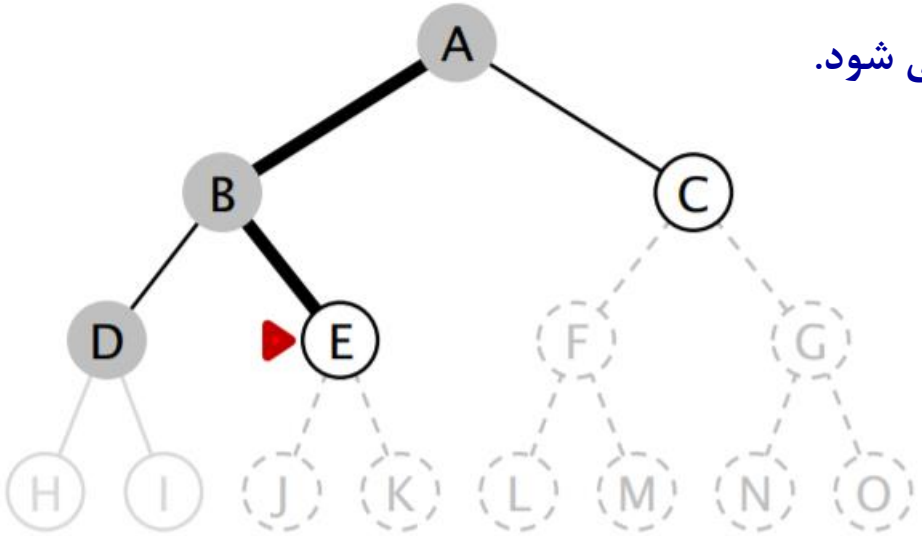


گره **H** چون فرزندی ندارد از حافظه حذف می شود.

Fringe=[I,E,C]

جستجوهای ناآگاهانه

- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.

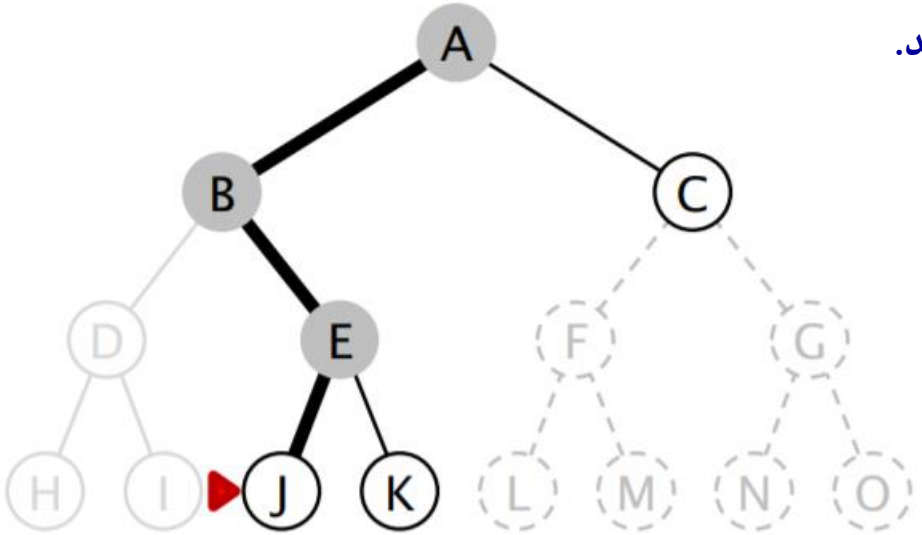


گره I چون فرزندی ندارد از حافظه حذف می شود.

Fringe=[E,C]

جستجوهای ناآگاهانه

- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.

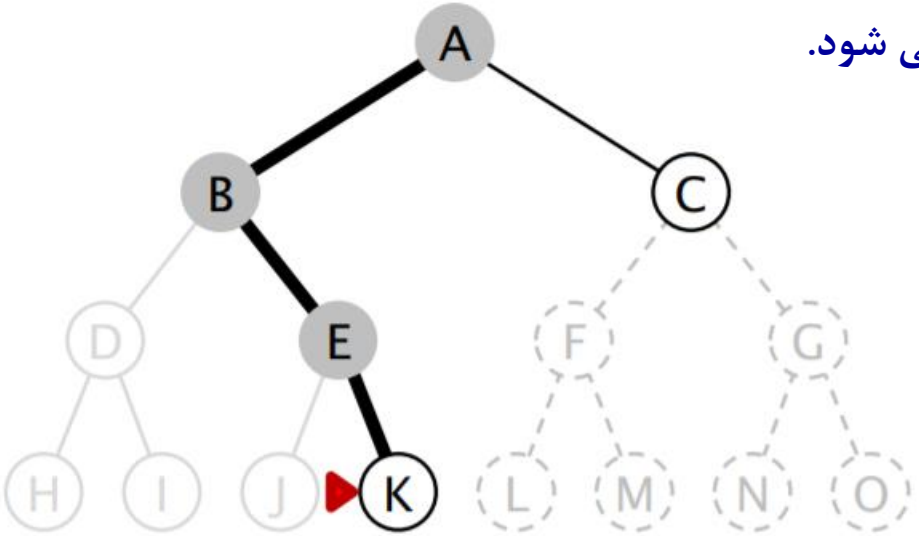


گره D و فرزندانش از حافظه حذف می شوند.

Fringe=[J,K,C]

جستجوهای ناآگاهانه

- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.

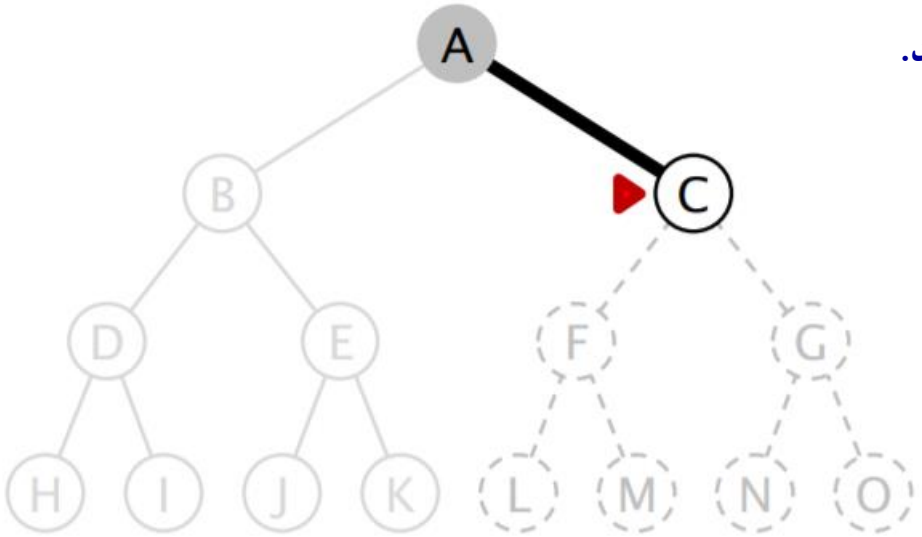


گره J چون فرزندی ندارد از حافظه حذف می شود.

Fringe=[K,C]

جستجوهای ناآگاهانه

- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.

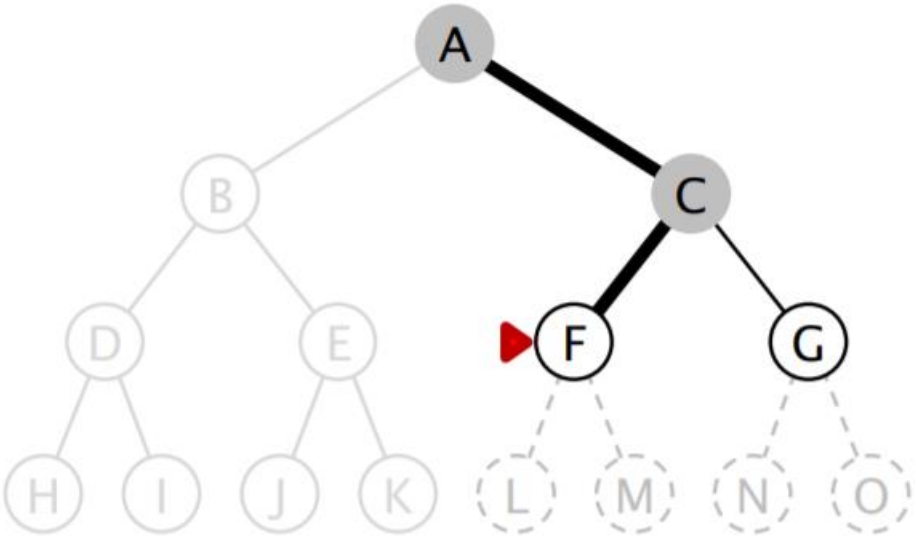


گره B و فرزندانش از حافظه حذف می شوند.

Fringe=[C]

جستجوهای ناآگاهانه

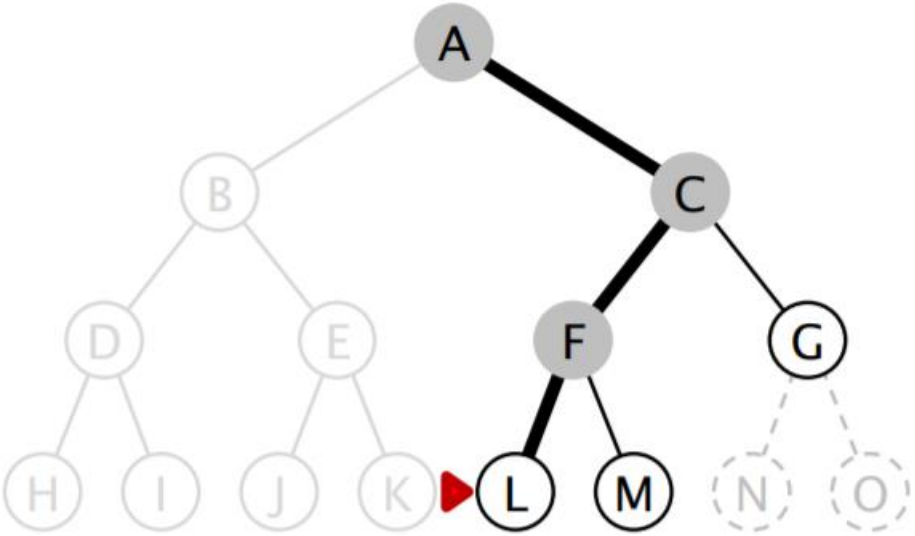
- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



Fringe=[F,G]

جستجوهای ناآگاهانه

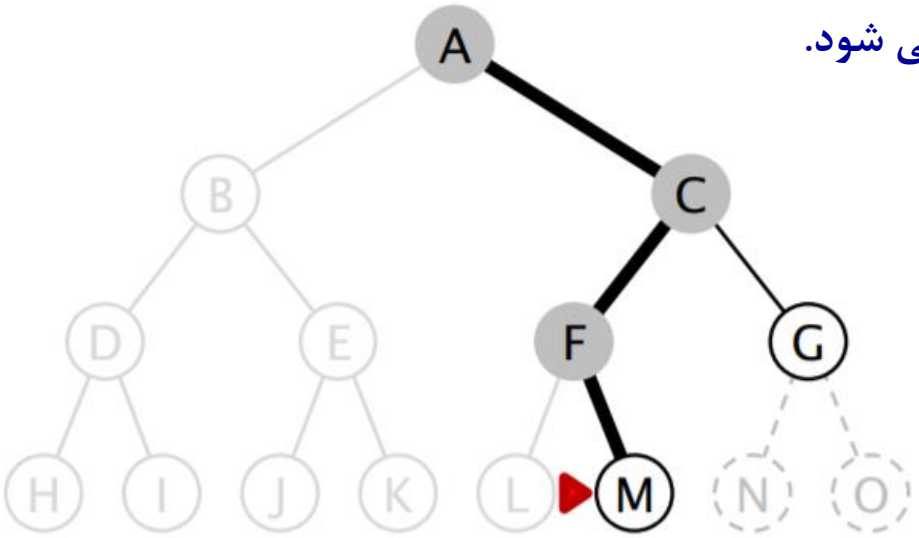
- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



Fringe=[L,M,G]

جستجوهای ناآگاهانه

- جستجوی عمقی. هر بار عمیق‌ترین گره گسترش نیافته را گسترش می‌دهد.
- پیاده‌سازی. با استفاده از پشته؛ درج فرزندان جدید در ابتدای لیست.



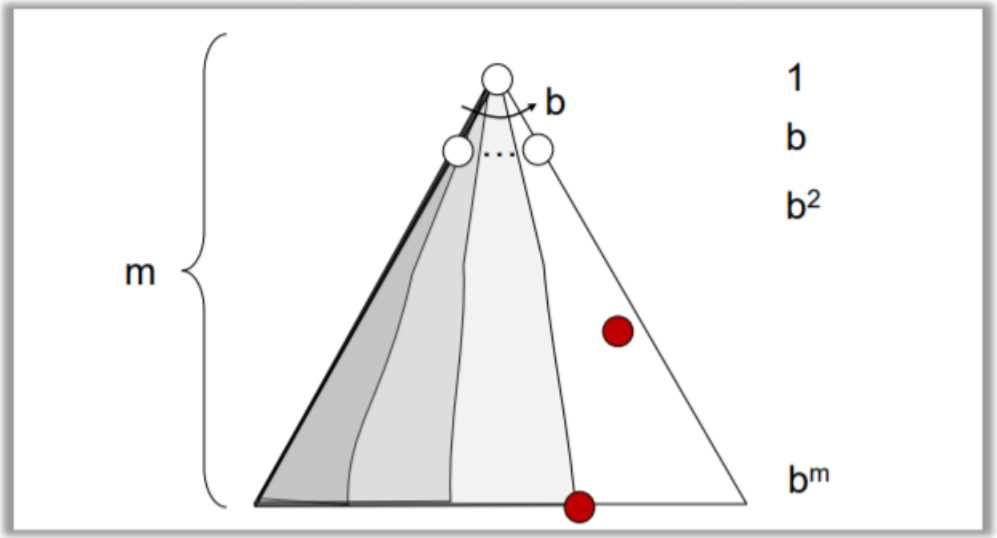
گره L چون فرزندی ندارد از حافظه حذف می شود.

Fringe=[M,G]

جستجوهای ناآگاهانه

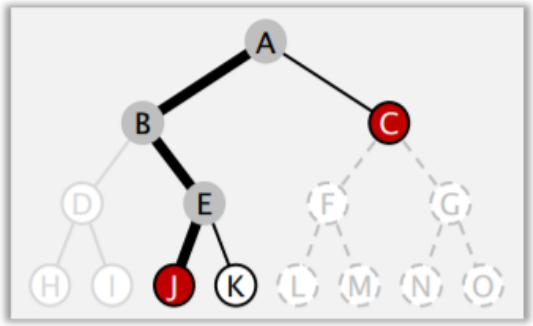
ارزیابی جستجوی عمقی

- جستجوی عمقی کدام گره‌ها را گسترش می‌دهد؟
  - تمام گره‌های قبل از چپ‌ترین گره راه‌حل.



جستجوهای ناآگاهانه

ارزیابی جستجوی عمقی



- کامل؟
  - جستجوی درختی: خیر
  - جستجوی گرافی: بله [فضای حالت متناهی]
- بهینه؟ خیر (شکل روبرو)
- پیچیدگی زمانی؟ نمایی

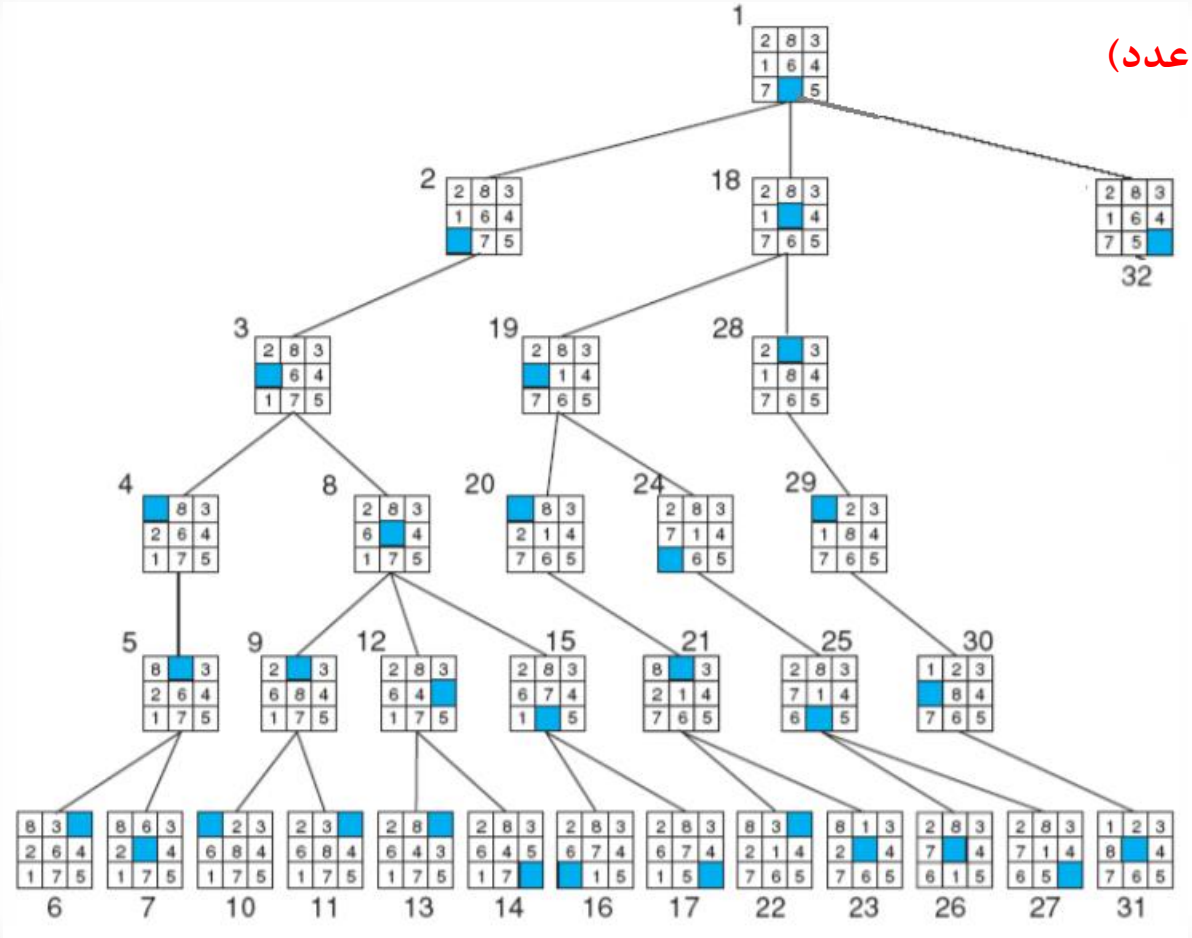
$$b^1 + b^2 + b^3 + \cdots + b^m \in O(b^m)$$

- پیچیدگی حافظه؟ خطی [جستجوی درختی]

$$O(bm)$$

جستجوهای ناآگاهانه

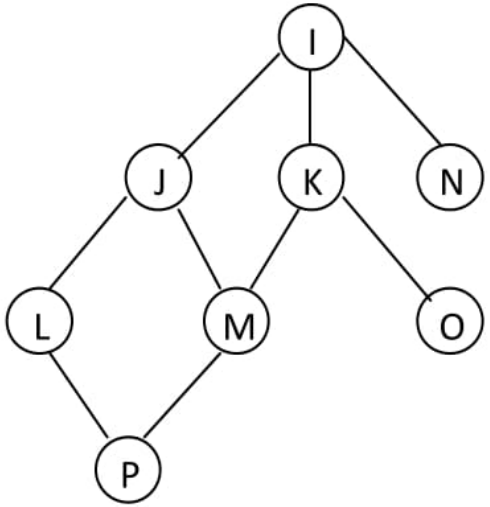
مثال: جستجوی عمقی (جورچین ۸ عدد)



جستجوهای ناآگاهانه

تمرین:

اگر در نمودار شکل زیر جستجوی اول عمق را از گره K آغاز کنیم، کدام گره ها به ترتیب از چپ به راست گسترش می یابند؟ (فرض کنید فرزندان یک گره بر اساس ترتیب حروف الفبا انتخاب می شوند.)



۱. K،I،J،L، M،P،N،O

۲. K،O،J،L،P،M،I،N

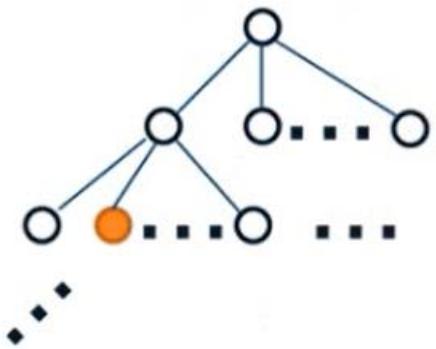
۳. K،I،J،L،P،M،N،O

۴. K،I،N،J،L،M،P،O

جستجوهای ناآگاهانه

تمرین:

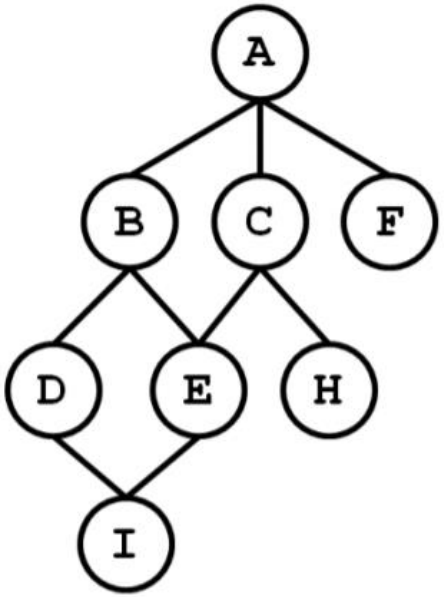
اگرفاکتورانشعاب یک درخت جستجوb باشد،وهدف در دومین‌گره ازچپ در عمق۲ قرار داشته باشد درصورت استفاده از هرکدام از الگوریتم‌های DFSوBFSچندگره باید بررسی شود،تا هدف پیدا شود.(فرض کنید تست هدف در لحظه بسط هدف انجام می‌شود، و ریشه درعمق صفر است،وارتفاع درخت۴ است)



جستجوهای ناآگاهانه

تمرین:

اگر در گراف روبرو جستجوی اول عمق (DFS) را از راس C شروع کنیم، به ترتیب کدام گره‌ها دیده می‌شوند؟ (فرض کنید فرزندان یک گره بر اساس ترتیب حروف الفبا انتخاب می‌شوند)

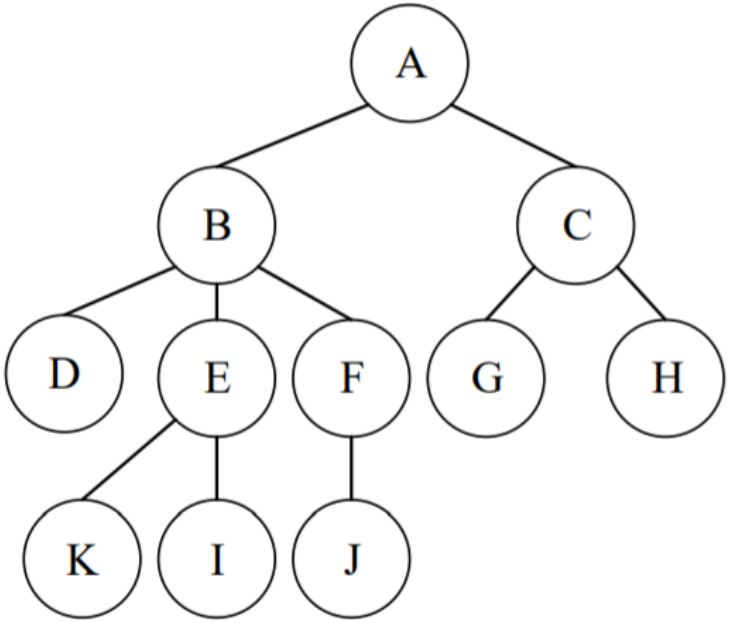


1. ABCDEFHI
2. CABDIEFH
3. CAEHBFID
4. CABDEHIF

جستجوهای ناآگاهانه

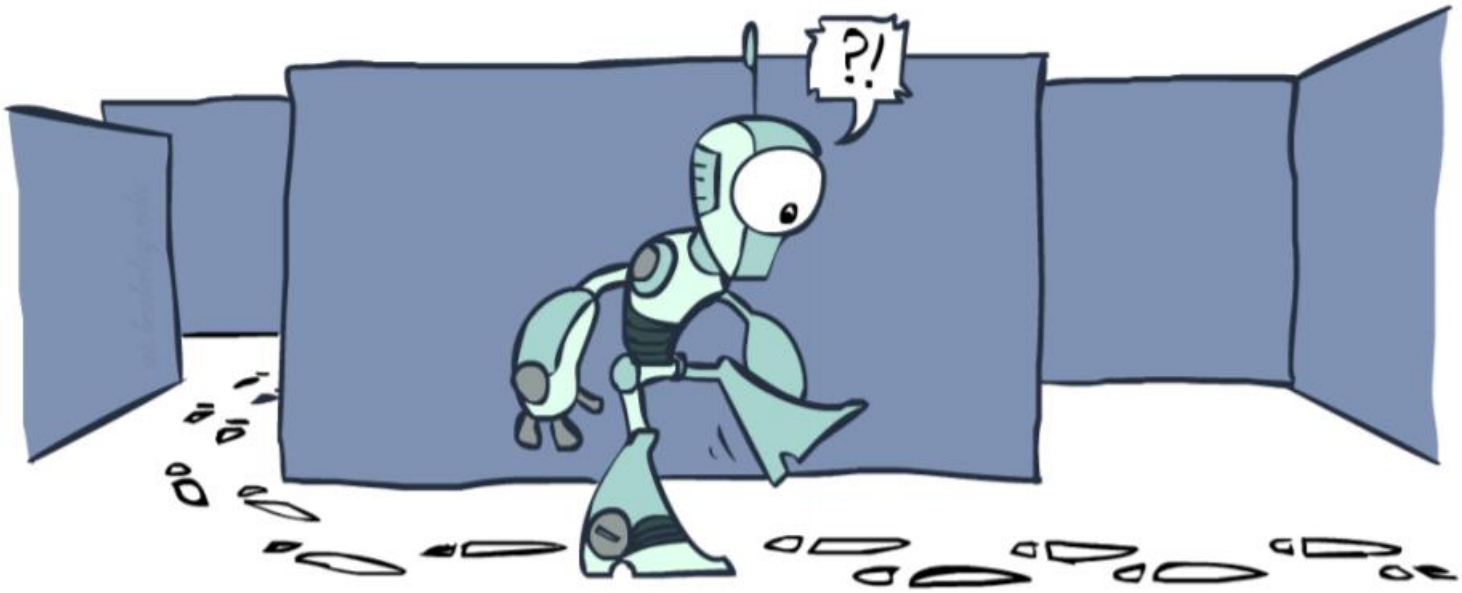
تمرین:

اگر گراف فضای حالت به صورت زیر باشد و A حالت شروع و H حالت هدف باشد با استفاده ار جستجوی DFS را ه حل را پیدا کنید ؟



جستجوهای ناآگاهانه

جستجوی گرافی

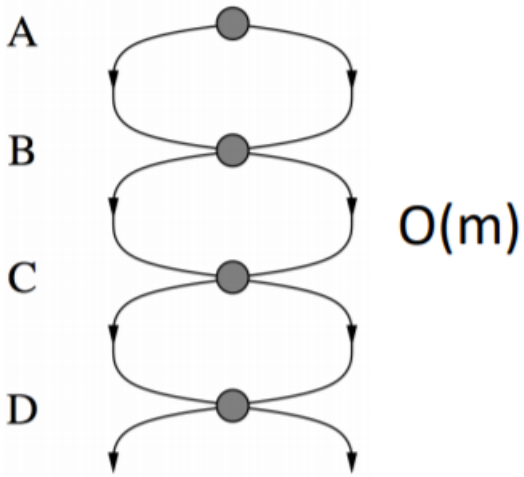


جستجوهای ناآگاهانه

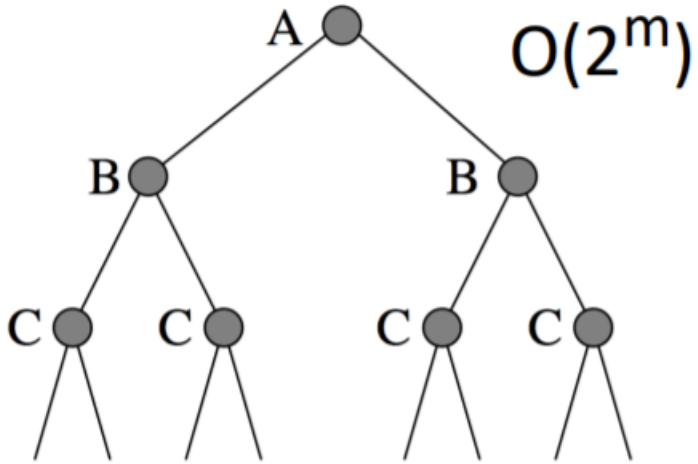
جستجوی گرافی

- شکست در تشخیص حالت‌های تکرای می‌تواند یک مسئله‌ی خطی را به یک مسئله‌ی نمایی تبدیل کند!

گراف فضای حالت



درخت جستجو



جستجوهای ناآگاهانه

جستجوی گرافی

- ایده. اجتناب از گسترش دادن حالت‌های تکراری.
- پیاده‌سازی.
  - جستجوی درختی به همراه مجموعه گره‌های گسترش یافته
  - درخت جستجو را گره به گره گسترش بده، اما ...
  - قبل از گسترش دادن یک گره، بررسی کن که حالت آن گره یک حالت جدید باشد
  - در غیر این صورت گره بعدی را انتخاب کن
- مهم. حالت گره‌های گسترش یافته در یک مجموعه ذخیره می‌شوند و نه در یک لیست.
- س. آیا جستجوی گرافی ممکن است باعث ناکامل شدن یک استراتژی جستجو گردد؟
- س. آیا جستجوی گرافی ممکن است باعث غیربهینه شدن یک استراتژی جستجو گردد؟

جستجوهای ناآگاهانه

الگوریتم جستجوی گرافی

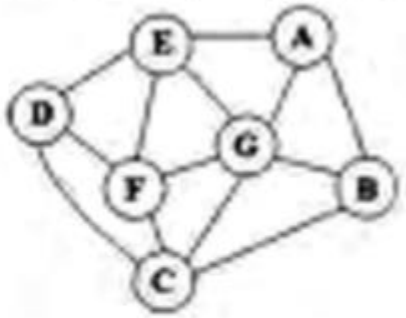
```
function GRAPH-SEARCH(problem) returns a solution, or failure
  initialize the frontier using the initial state of problem
  initialize the explored set to be empty
  loop do
    if the frontier is empty then return failure
    choose a leaf node and remove it from the frontier
    if the node contains a goal state then return the corresponding solution
    add the node to the explored set
    expand the chosen node, adding the resulting nodes to the frontier
      only if not in the frontier or explored set
  end
```

- ایده‌ی جستجوی گرافی.
  - ذخیره کردن گره‌های گسترش یافته.
  - جلوگیری از گسترش دادن گره‌هایی که قبلاً گسترش یافته‌اند.

جستجوهای ناآگاهانه

تمرین:

در گراف زیر با انجام جستجوی اول عمق و شروع از راس D، به روش جستجوی درختی (Tree Search) و با جستجوی گرافی (Graph Search) کدام گره ها به ترتیب از چپ به راست گسترش می یابند؟ (فرزندان یک گره را به ترتیب حروف الفبا انتخاب کنید.)



۱. با جستجوی درختی: DCBAEFG و با جستجوی گراف: DCEFBGA

۲. با جستجوی درختی: DCBAEFG و با جستجوی گراف: DCBAGEF

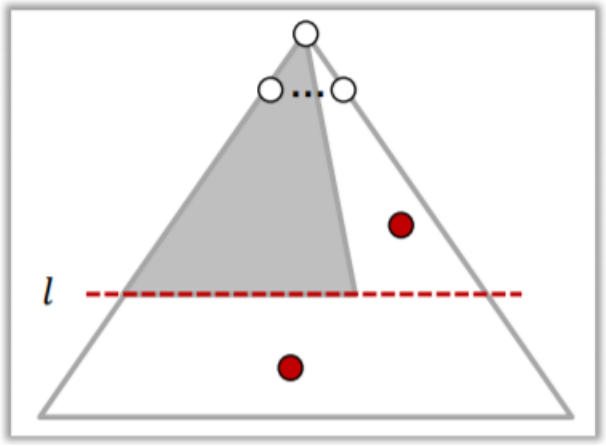
۳. با جستجوی درختی: DCEFBGA و با جستجوی گراف: DCBGAEF

۴. با جستجوی درختی: DCBGAEF و با جستجوی گراف: DCEFBGA

جستجوهای ناآگاهانه

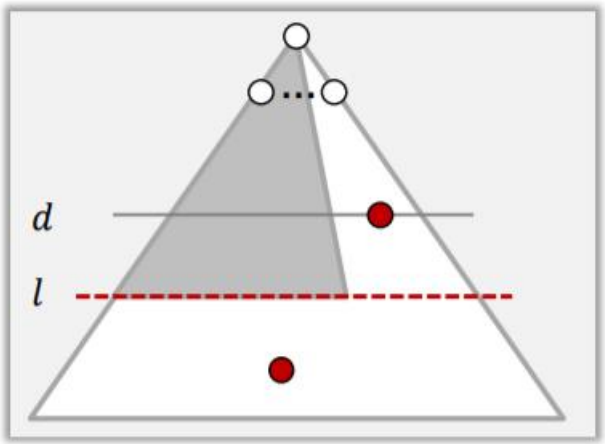
جستجوی عمقی محدود

- ایده. محدود کردن عمق جستجو در جستجوی عمقی.
  - فرزندان گره‌های واقع در عمق l تولید نمی‌شوند.



جستجوهای ناآگاهانه

ارزیابی جستجوی عمقی محدود



- کامل؟ بله [اگر $l \geq d$]
- بهینه؟ خیر (همانند جستجوی عمقی)
- پیچیدگی زمانی؟ نمایی

$$b^1 + b^2 + b^3 + \cdots + b^l \in O(b^l)$$

- پیچیدگی حافظه؟ خطی [جستجوی درختی]

$$O(bl)$$

جستجوهای ناآگاهانه

شبه کد جستجوی عمقی محدود

```
function DEPTH-LIMITED-SEARCH(problem, limit) returns a solution, or failure/cutoff
  return RECURSIVE-DLS(MAKE-NODE(problem.INITIAL-STATE), problem, limit)

function RECURSIVE-DLS(node, problem, limit) returns a solution, or failure/cutoff
  if problem.GOAL-TEST(node.STATE) then return SOLUTION(node)
  else if limit = 0 return cutoff
  else
    cutoff_occured? ← false
    for each action in problem.ACTIONS(node.STATE) do
      child ←CHILD-NODE(problem, node, action)
      result ← RECURSIVE-DLS(child, problem, limit − 1)
      if result = cutoff then cutoff_occured? ← true
      else if result ≠ failure then return result
    if cutoff_occured? then return cutoff else return failure
```

ایراد اصلی. تعیین یک محدوده‌ی عمقی مناسب!

Cutoff همان L است. برای تعیین L در برخی مسایل مثل (آراد - بخارست) می توانیم L را بزرگتر از d انتخاب کنیم.
یعنی بدترین حالت ممکن آن است که برای رسیدن از شهر آراد به شهر بخارست از همه شهرهای رومانی عبور کنیم.
ولی در بازی شطرنج تعیین محدوده برای عمق امکان پذیر نیست.

جستجوهای ناآگاهانه

۵) جستجوی عمقی تکراری

- یک استراتژی برای یافتن بهترین مقدار L
- روشی برای ترکیب مزایای جستجوی BF و DF

function ITERATIVE_DEEPENING_SEARCH(*problem*) **return** a solution or failure

inputs: *problem*

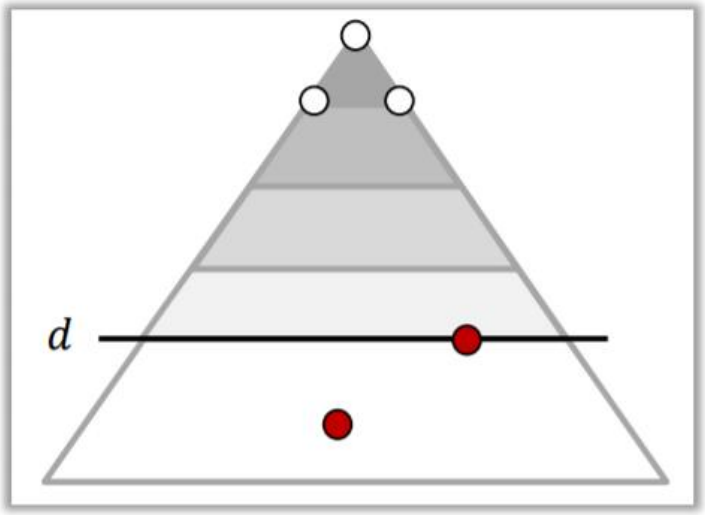
for *depth* ← 0 to ∞ **do**

result ← DEPTH-LIMITED_SEARCH(*problem*, *depth*)

if *result* ≠ *cuttoff* **then return** *result*

جستجوهای ناآگاهانه

- ایده. ترکیب مزایای جستجوی سطحی و جستجوی عمقی.
  - یک جستجوی عمقی با محدوده‌ی عمقی ۰ انجام بده. اگر راه‌حل پیدا نشد ...
  - یک جستجوی عمقی با محدوده‌ی عمقی ۱ انجام بده. اگر راه‌حل پیدا نشد ...
  - یک جستجوی عمقی با محدوده‌ی عمقی ۲ انجام بده. اگر راه‌حل پیدا نشد ...
  - یک جستجوی عمقی با محدوده‌ی عمقی ۳ انجام بده. اگر راه‌حل پیدا نشد ...
  - ...

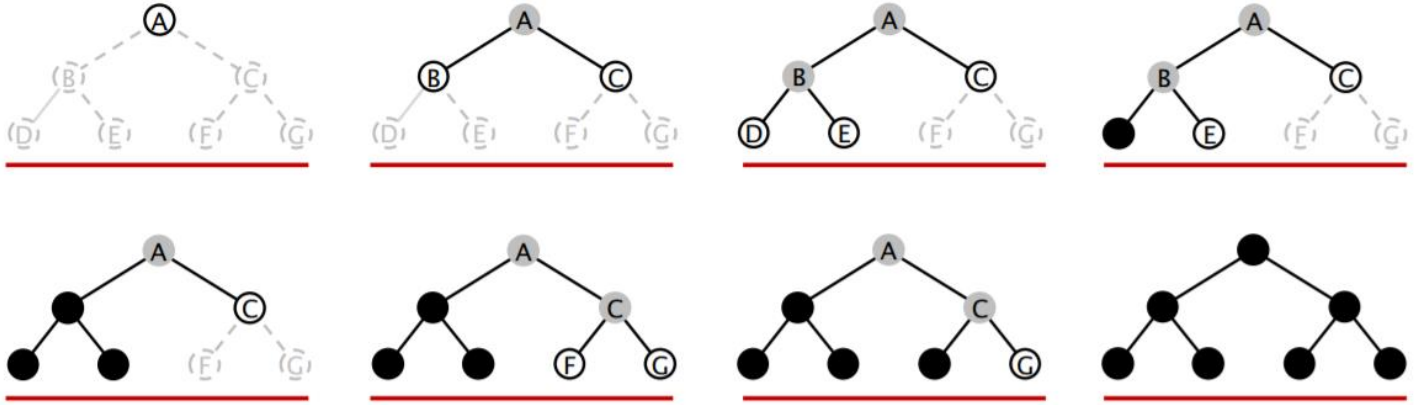


جستجوهای ناآگاهانه

- تکرار اول. $l = 0$

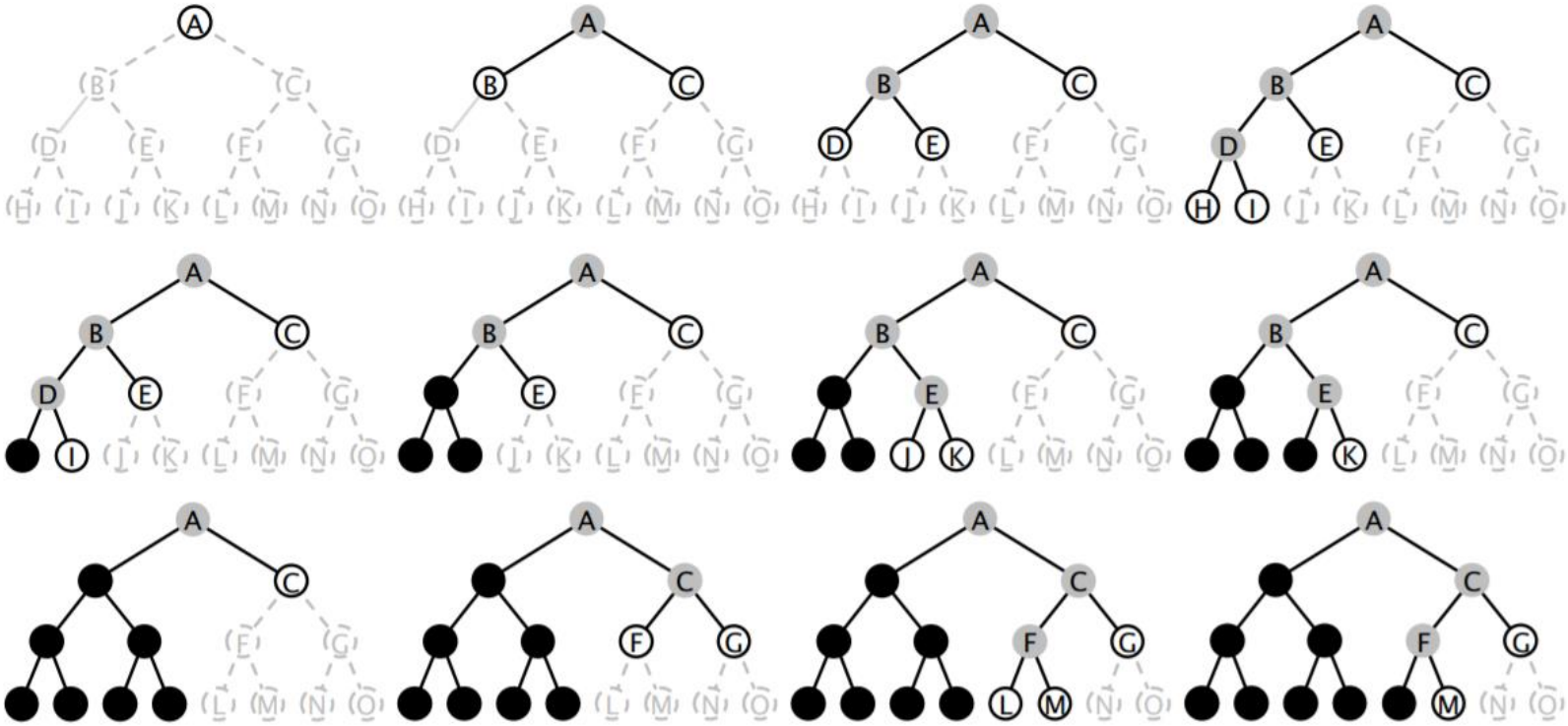
- تکرار دوم. $l = 1$

- تکرار سوم. $l = 2$



جستجوهای ناآگاهانه

- تکرار چهارم. $l = 3$



جستجوهای ناآگاهانه

ارزیابی جستجوی عمقی تکراری

- کامل؟ بله [مانند جستجوی سطحی]
- بهینه؟ بله [مانند جستجوی سطحی]
- پیچیدگی زمانی؟ نمایی [مانند جستجوی عمقی] $b^1 + b^2 + b^3 + \cdots + b^l \in O(b^l)$
- پیچیدگی حافظه؟ خطی [مانند جستجوی عمقی] $O(bd)$

نکته: این الگوریتم بهترین الگوریتم از نوع جستجوی نا آگاهانه است.

جستجوهای ناآگاهانه

جستجوی عمقی تکراری

- پیچیدگی زمانی؟ $O(b^d)$

- level d: once
- level d-1: 2
- level d-2: 3
- …
- level 2: d-1
- level 1: d
- Level 0: d+1

$$N(IDS) = (d)b + (d-1)b^2 + \ldots + (1)b^d$$

نکته: در شمارش گره‌های تولید شده در هر کدام از روش‌های گفته شده گره مبدا را در نظر نمی‌گیریم.

- پیچیدگی حافظه؟ $O(bd)$

جستجوهای ناآگاهانه

جستجوی عمقی تکراری

- مثال: تعداد گره های تولید شده با جستجوی سطری و عمقی تکراری در یک مساله فرضی با فاکتور انشعاب ۱۰ و عمق جواب بهینه ۵ را بدست آورید.

$$N(IDS) = (d)b + (d-1)b^2 + \ldots + (1)b^d$$
$$N(IDS) = 50 + 400 + 3000 + 20000 + 100000 = 123450$$

$$N(BFS) = b + b^2 + \ldots + b^d + (b^{d+1} - b)$$
$$N(BFS) = 10 + 100 + 1000 + 10000 + 100000 + 999990 = 1111100$$

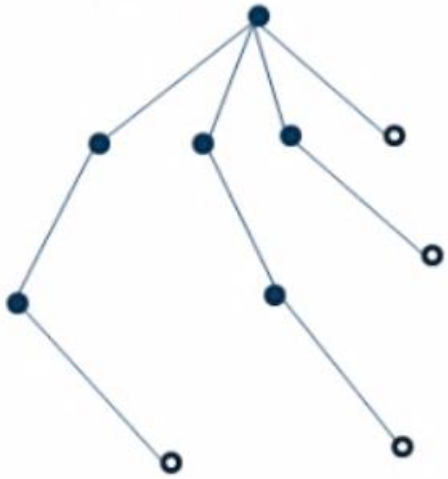
- تعداد گره های بسط داده شده؟

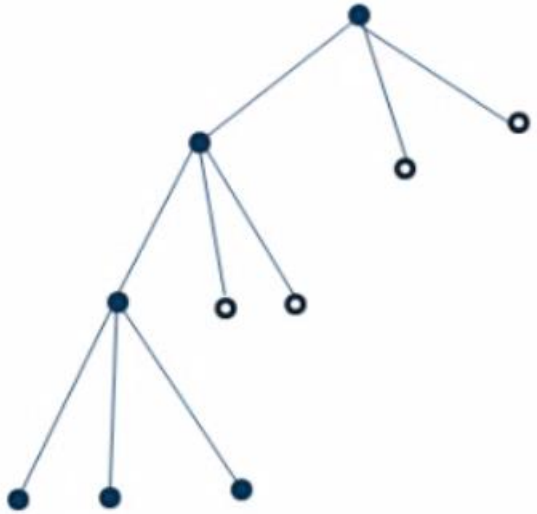
نکته: اگرفقط تعداد گره های بسط داده شده مدنظر باشد در BFS از سطر آخر صرف نظر می کنیم چون آنها فقط گره های تولید شده هستند. در IDS تولید شده وبسط یافته یکی هستند.

جستجوهای ناآگاهانه

تمرین:

اگر درحین عمل جستجو درخت جستجو،به شکل زیر گسترش بیابد.بنظرشما ازکدام روش استفاده شده است؟

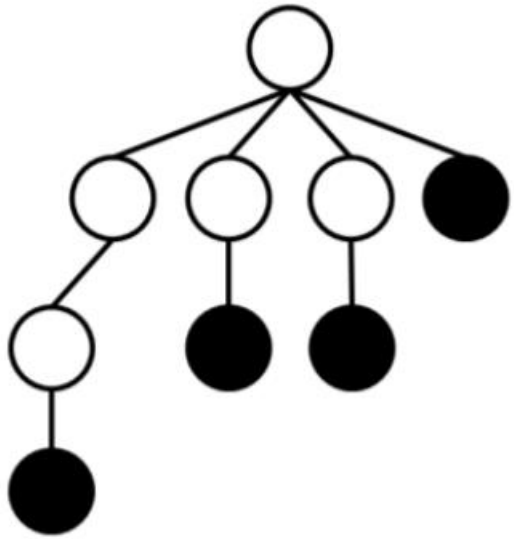




جستجوهای ناآگاهانه

تمرین:

در حین انجام یک روش جستجو، درخت جستجوی حاصل به شکل مقابل رشد یافته است. راس‌هایی که نامزد بسط داده شدن هستند به رنگ سیاه مشخص شده‌اند. این جستجو چه روشی می‌تواند باشد؟



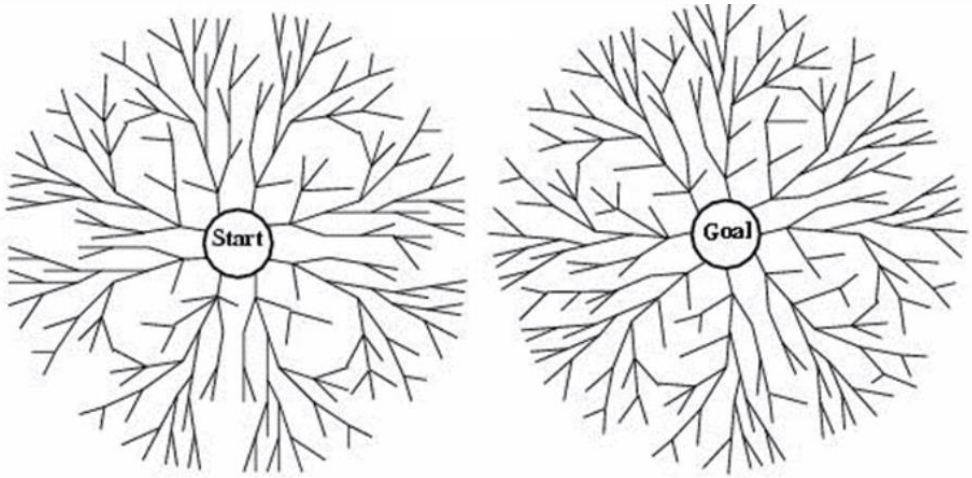
1. عمق نخست (dfs)
2. عرض نخست (bfs)
3. هزینه یکنواخت (ucs)
4. تعمیق تکراری (ids)

جستجوهای ناآگاهانه

جستجوی دوطرفه(Bidirectional Search)

- در جستجوی دوطرفه بطور همزمان،ازحالت شروع رو به جلو و از حالت هدف رو به عقب حرکت انجام می‌شود.
- پیچیدگی زمانی و حافظه این الگوریتم $O(b^{d/2})$ است.
- اگر چندین هدف مشخص وجود داشته باشد،میتوان مجموعه ای از حالات هدف را بصورت یک در نظر بگیریم.
- در حالتی‌که هدف غیرصریح باشد،استفاده ازجستجوی دوطرفه بصورت بهینه قابل اعمال نیست.

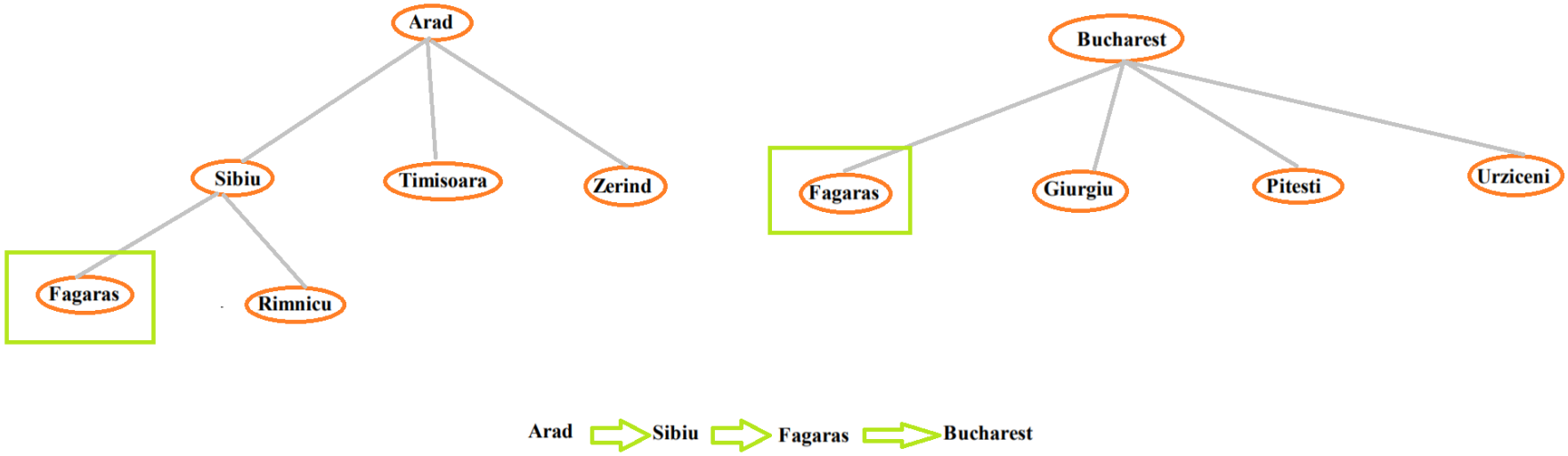
جستجوهای ناآگاهانه



- کامل بودن؟ بلی
  - ✓ به شرطی که از جستجوی سطری استفاده شود.
- بهینه بودن؟ بلی
  - ✓ به شرطی که از جستجوی سطری استفاده شود.

جستجوهای ناآگاهانه

مثال: جستجوی دو طرفه (BFS) مسیر آراد به بخارست



جستجوهای ناآگاهانه

جمع‌بندی جستجوهای ناآگاهانه

Criterion	Breadth-First	Uniform-cost	Depth-First	Depth-limited	Iterative deepening	Bidirectional search
Complete?	YES*	YES*	NO	YES, if $l \geq d$	YES	YES*
Time	b^{d+1}	$b^{C^*/e}$	b^m	b^l	b^d	$b^{d/2}$
Space	b^{d+1}	$b^{C^*/e}$	bm	bl	bd	$b^{d/2}$
Optimal?	YES*	YES*	NO	NO	YES	YES

جستجوهای ناآگاهانه

پایان فصل سوم