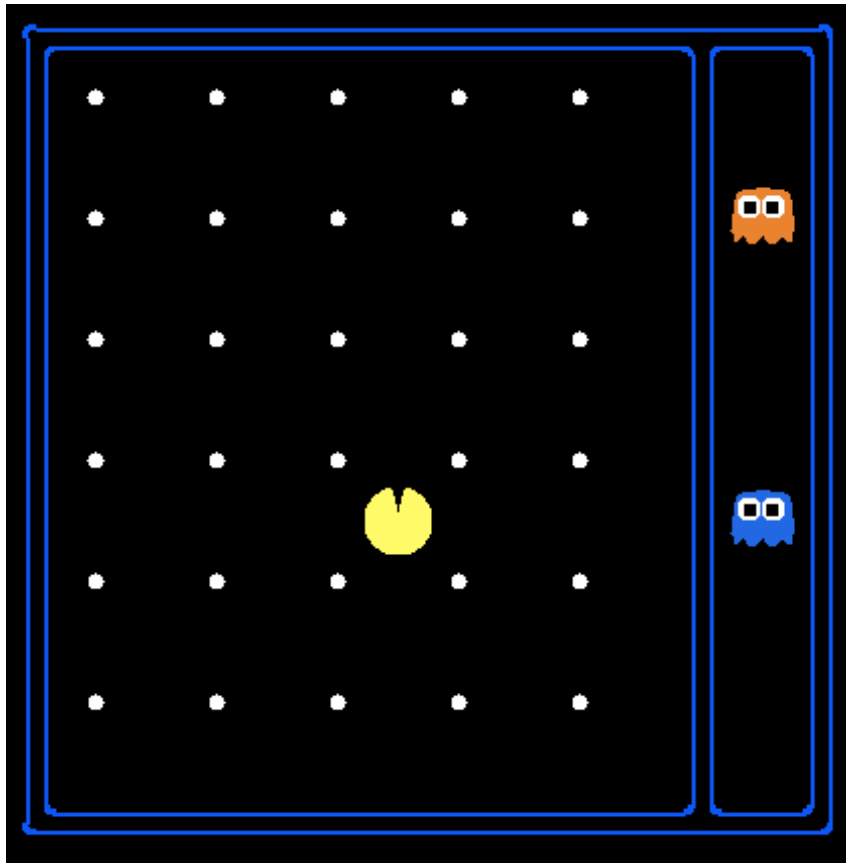


مسائل جستجو



اندازه فضای حالت



□ فضای حالت:

- مکان‌های ممکن برای عامل: ۱۲۰
- جهت‌های ممکن برای عامل: ۴
- حالت‌های ممکن برای هر کدام از ارواح: ۱۲
- تعداد نقاط: ۳۰

□ اندازه فضای جستجو اگر مسئله شامل تمام متغیرهای ممکن (مکان عامل، وضعیت نقاط، وضعیت ارواح و جهت حرکت) باشد؟

$$120 \times (2^{30}) \times (12^2) \times 4$$

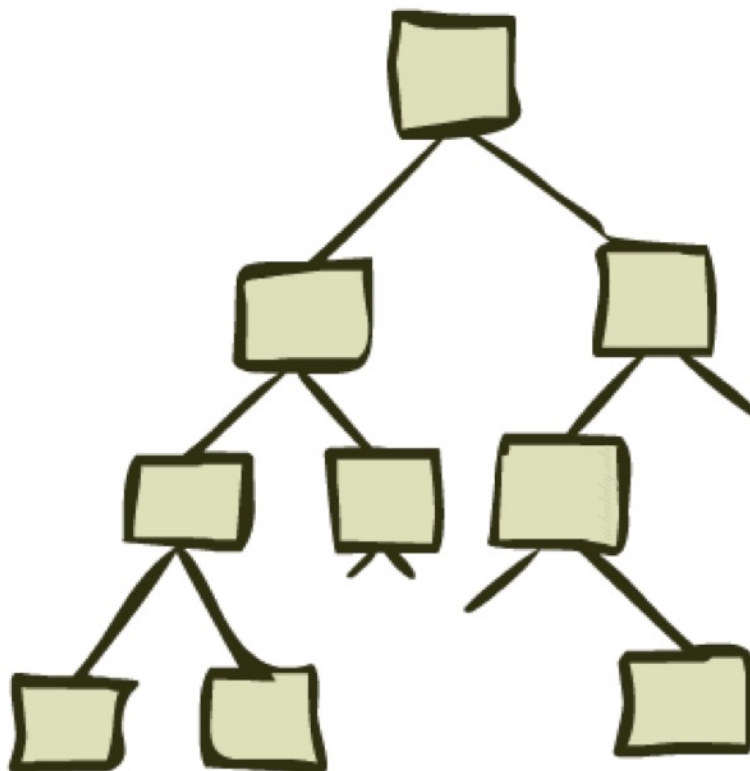
- تعداد حالت‌ها اگر مسئله مسیریابی باشد؟

$$120$$

- تعداد حالت‌ها اگر مسئله خوردن تمام نقاط باشد؟

$$120 \times (2^{30})$$

گراف فضای حالت و درخت جستجو



گراف فضای حالت

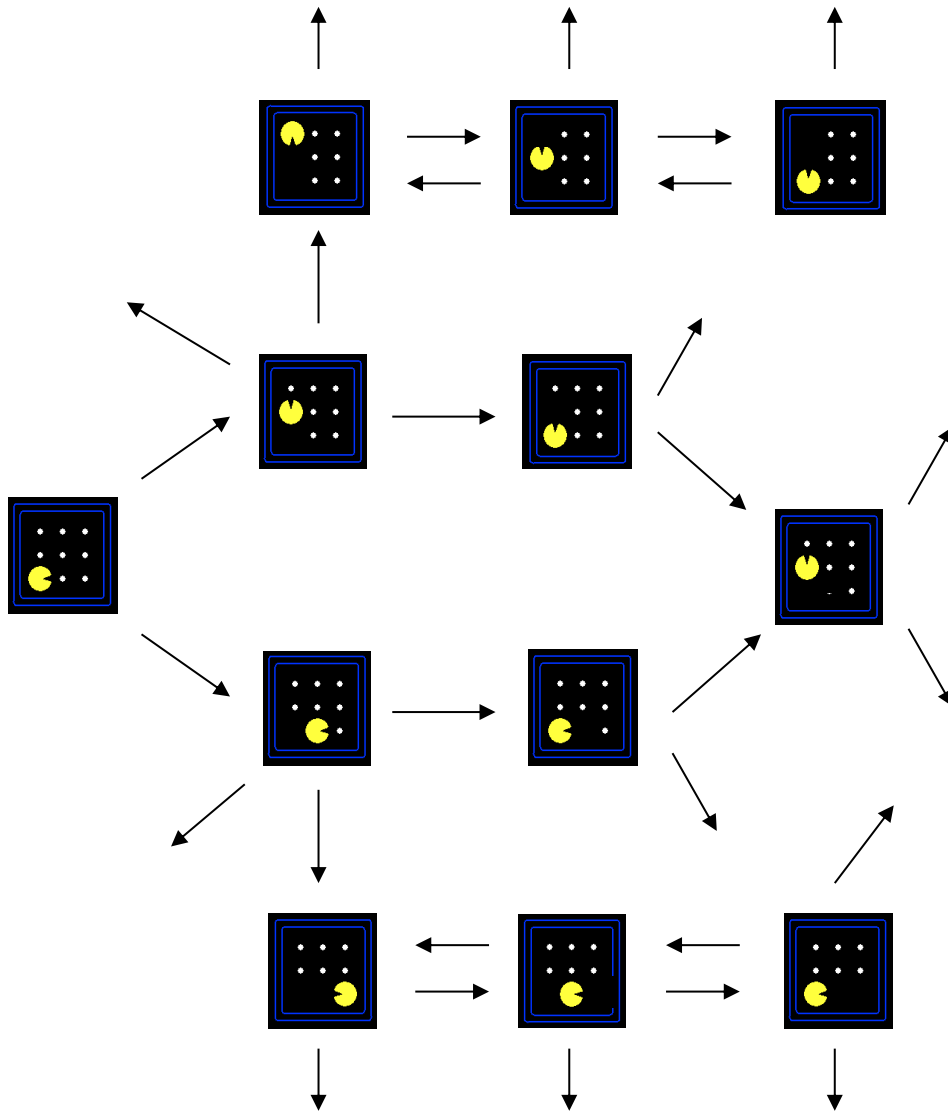
□ گراف فضای حالت:

همه‌ی حالات ممکن یک مسئله را نشان می‌دهد، و یال‌ها در آن نشان‌دهنده‌ی انتقال بین حالات هستند. (یک نمایش ریاضی از مسئله جستجو)

- رئوس: حالت‌های محیط (انتزاعی)
- یال‌ها: عملیات ممکن در هر رأس
- تابع حالت بعدی: رئوس همسایه
- حالت شروع: یکی از رئوس
- تست هدف: مجموعه‌ای از یک یا چند رأس

□ در گراف فضای حالت، رأس تکراری وجود ندارد!

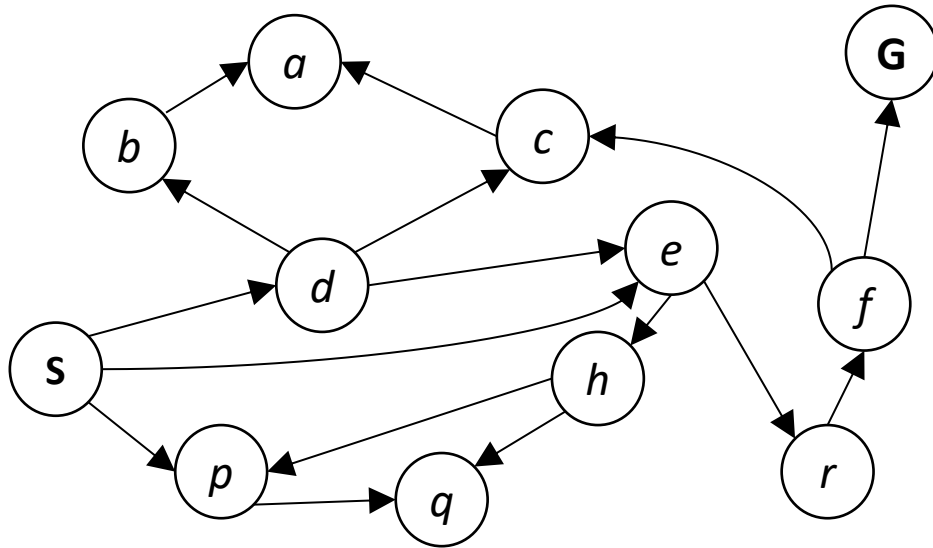
□ به‌ندرت می‌توان گراف کامل را در حافظه ساخت (چون خیلی بزرگ است)، اما همچنان یک مفهوم ارزشمند برای درک فضا محسوب می‌شود.



گراف فضای حالت

□ گراف فضای حالت: یک نمایش ریاضی از مسئله جستجو

- رئوس: حالت‌های محیط (انتزاعی)
- یال‌ها: عملیات ممکن در هر رأس
- تابع حالت بعدی: رئوس همسایه
- حالت شروع: یکی از رئوس
- تست هدف: مجموعه‌ای از یک یا چند رأس



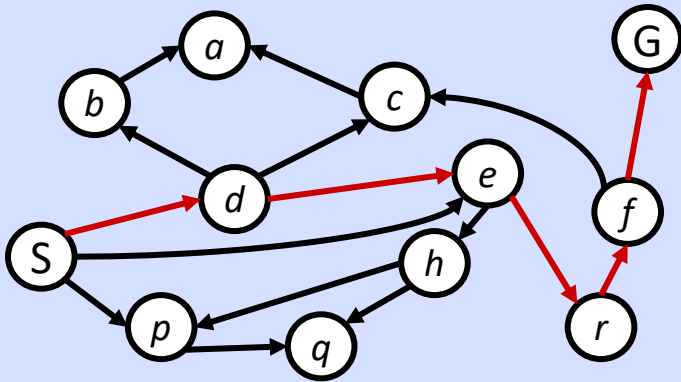
یک گراف فضای حالت کوچک که برای
یک مسئله جستجوی کوچک است

□ در گراف فضای حالت، رأس تکراری وجود ندارد!

□ به ندرت می‌توان گراف کامل را در حافظه ساخت (چون خیلی بزرگ است)، اما همچنان یک مفهوم ارزشمند برای درک فضا محسوب می‌شود.

مقایسه گراف فضای حالت و درخت جستجو

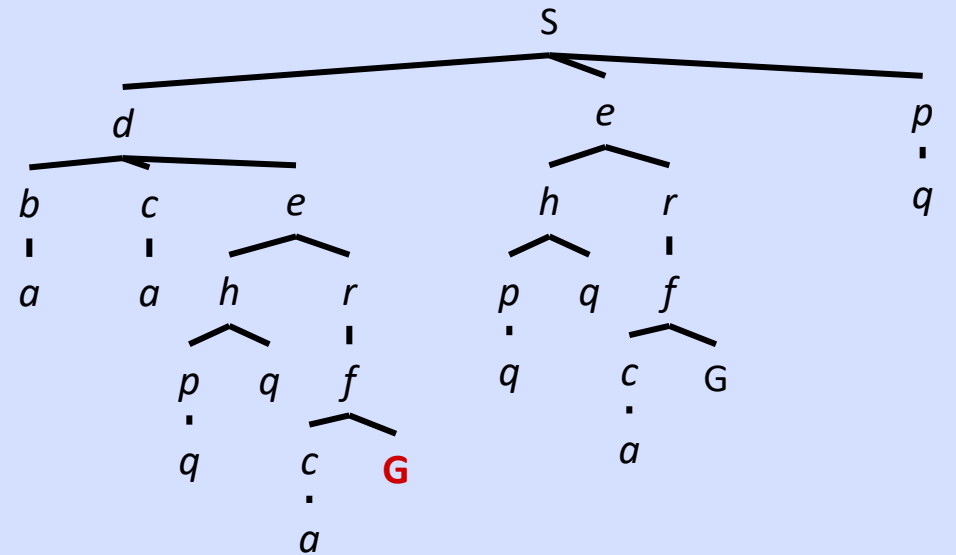
گراف فضای حالت



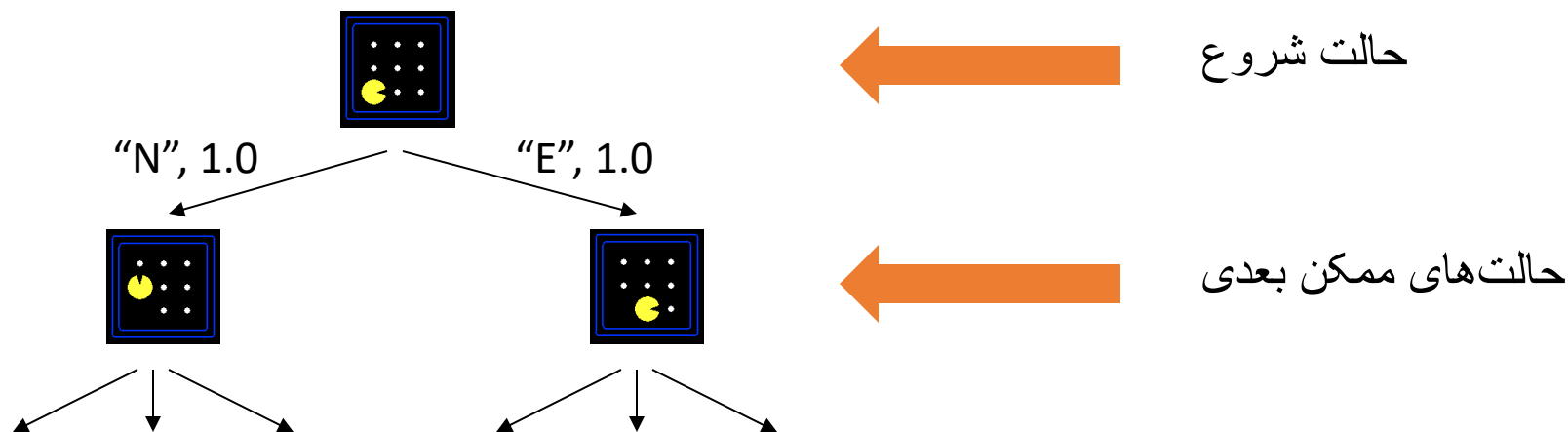
هر گره (NODE) در
درخت جستجو یک
مسیر کامل در گراف
فضای حالت است.

درخت را به اندازه‌ای که
نیاز است می‌سازیم.
لزومی ندارد درخت کامل
ساخته شود.

درخت جستجو



درخت جستجو



□ درخت جستجو

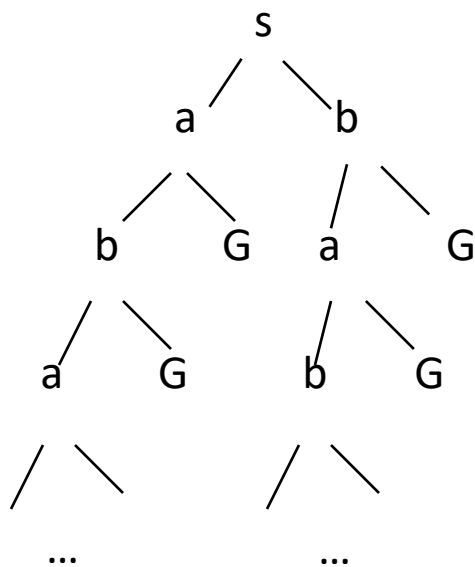
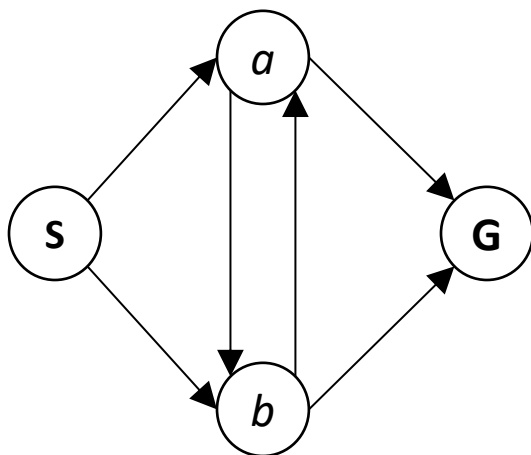
یک نمایش درختی از مسیرهای جستجو است که یک الگوریتم جستجو برای یافتن پاسخ ایجاد می‌کند.

- ریشه برابر با حالت شروع مسئله است.
- درخت از یک گره ریشه شروع شده و مسیرهای مختلف را بررسی می‌کند.
- هر گره فقط یک والد دارد
- در اغلب مسائل، هرگز نمی‌توان کل درخت را ساخت، زیرا بیش از حد بزرگ است.!!!

مقایسه گراف فضای حالت و درخت جستجو

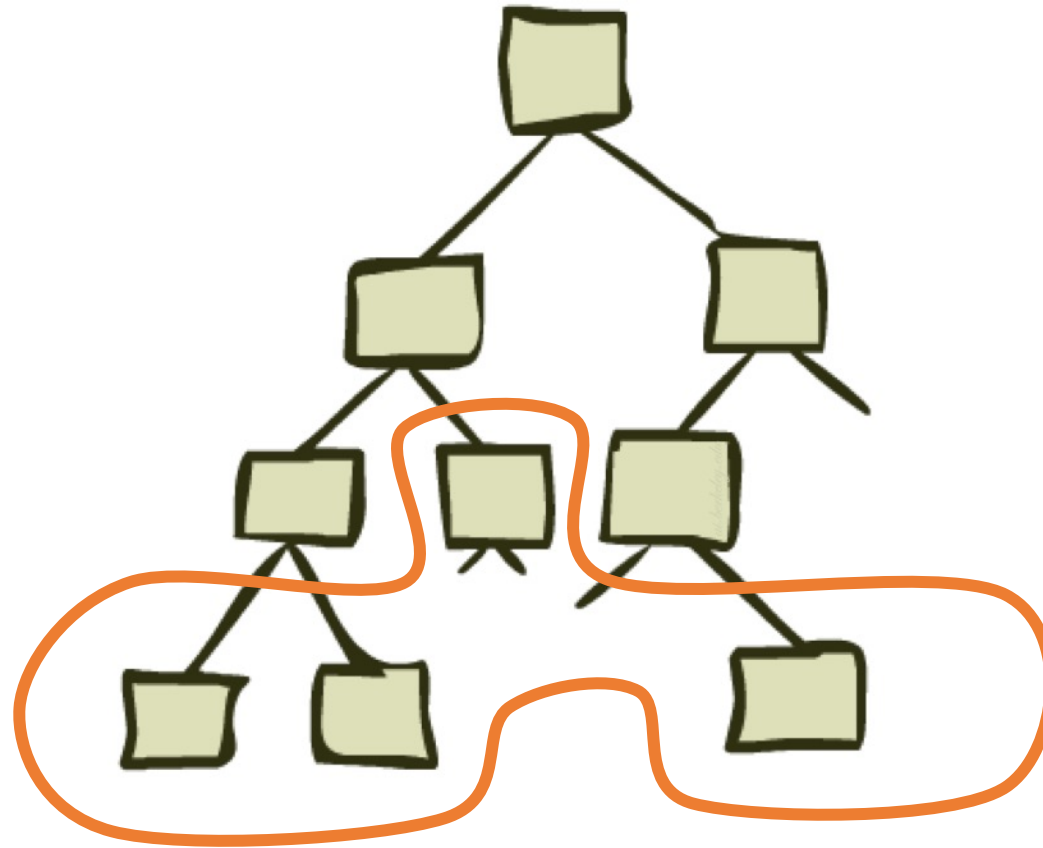
اندازه درخت جستجوی آن (با حالت شروع S) چقدر است؟

این گراف ۴-حالتی را در نظر بگیرید:

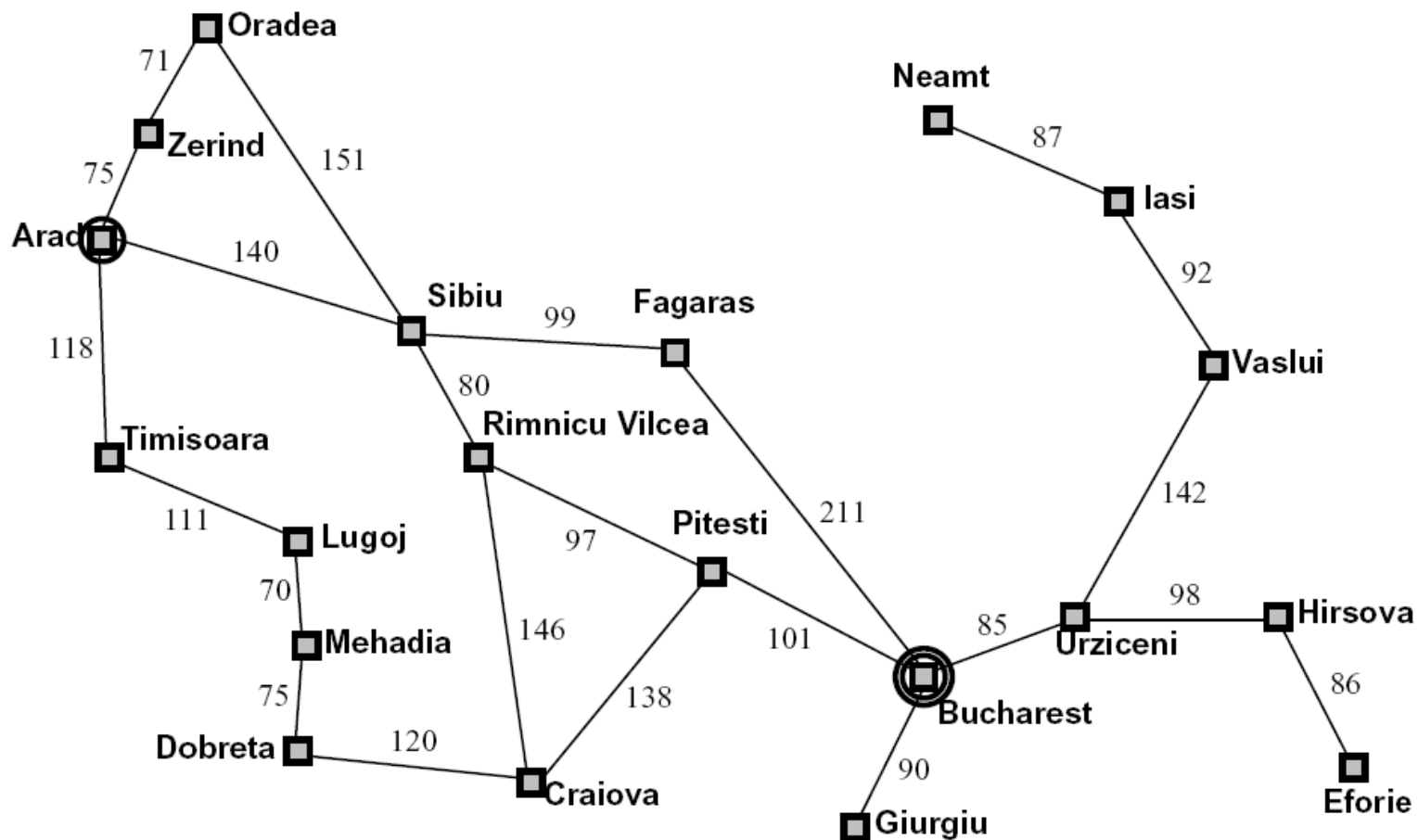


ساختارهای تکراری زیادی در درخت جستجو وجود دارد!

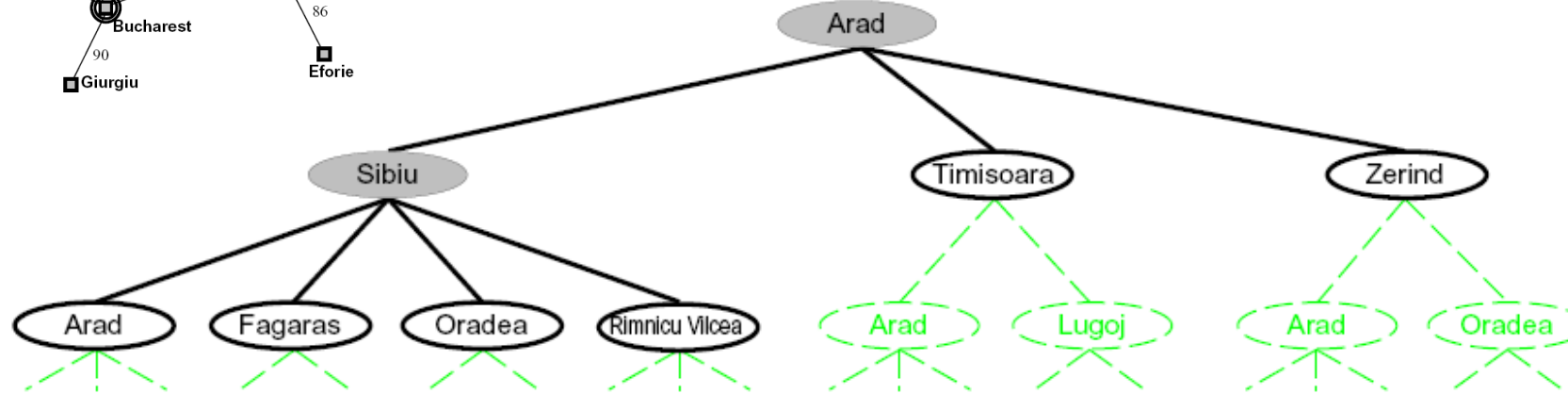
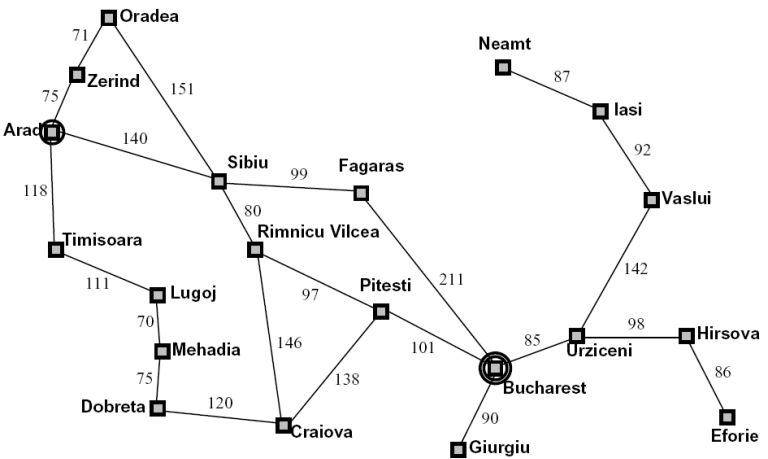
درخت جستجو



مثال جستجو: مسیریابی در رومانی



جستجوی فضای حالت با استفاده از درخت جستجو



□ جستجو:

- انتخاب یک گره
- تست هدف
- گسترش گره انتخاب شده

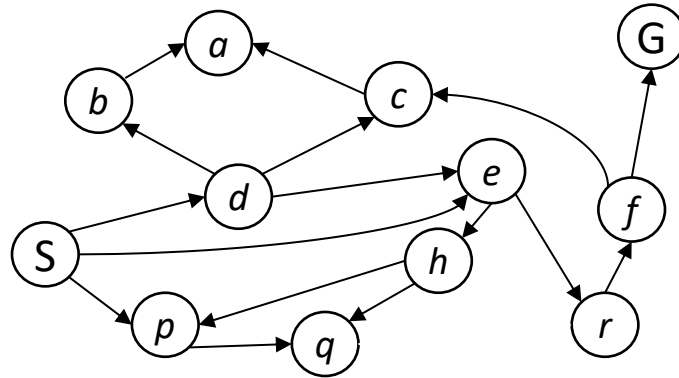
الگوریتم جستجوی عمومی درخت (General Tree Search)

```
function TREE-SEARCH(problem, strategy) returns a solution, or failure
  initialize the search tree using the initial state of problem
  loop do
    if there are no candidates for expansion then return failure
    choose a leaf node for expansion according to strategy
    if the node contains a goal state then return the corresponding solution
    else expand the node and add the resulting nodes to the search tree
  end
```

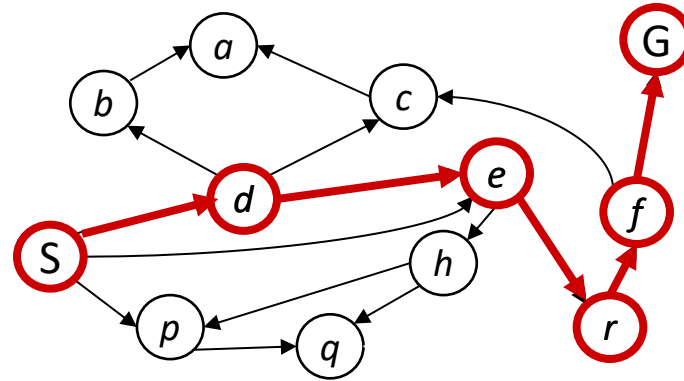
□ مفاهیم مهم:

- Fringe
 - مجموعه‌ای از گره‌هایی است که در طول فرآیند جستجو تولید شده‌اند ولی هنوز گسترش نیافته‌اند. (این گره‌ها در صف انتظار هستند تا پردازش شوند.)
- Expansion (گسترش گره‌ها)
 - فرآیند گسترش یک گره یعنی انتخاب یک گره از حاشیه و تولید جانشین‌های آن بر اساس action‌های موجود در مسئله. این گره دیگر گسترش یافته محسوب می‌شود و از حاشیه حذف می‌شود.
- Exploration Strategy (استراتژی جستجو)
 - کدام گره از حاشیه انتخاب شود تا گسترش یابد. (بر اساس نوع جستجو)

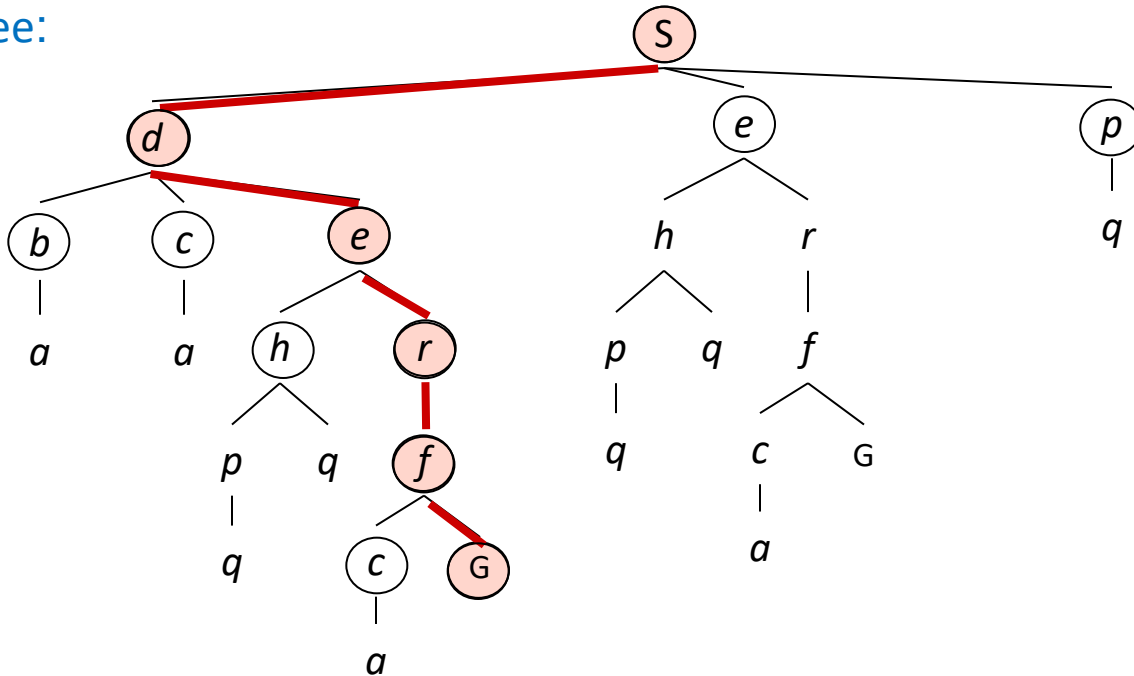
مثال: درخت جستجو



مثال: درخت جستجو



Tree:



Fringe:

~~S~~
~~S → d~~
 S → e
 S → p
 S → d → b
 S → d → c
~~S → d → e~~
 S → d → e → h
~~S → d → e → r~~
~~S → d → e → r → f~~
 S → d → e → r → f → c
~~S → d → e → r → f → G~~

روش‌های جستجوی ناآگاهانه

(Uninformed Search Methods)

1. جستجوی عمق اول (Depth-First Search - DFS)
2. جستجوی سطح اول (Breadth-First Search - BFS)
3. جستجوی عمقی افزایشی (ترکیب DFS و BFS) (Iterative Deepening Search - IDS)
4. جستجوی هزینه یکنواخت (Uniform-Cost Search - UCS)

روش‌های جستجوی ناآگاهانه (Uninformed Search Strategies)

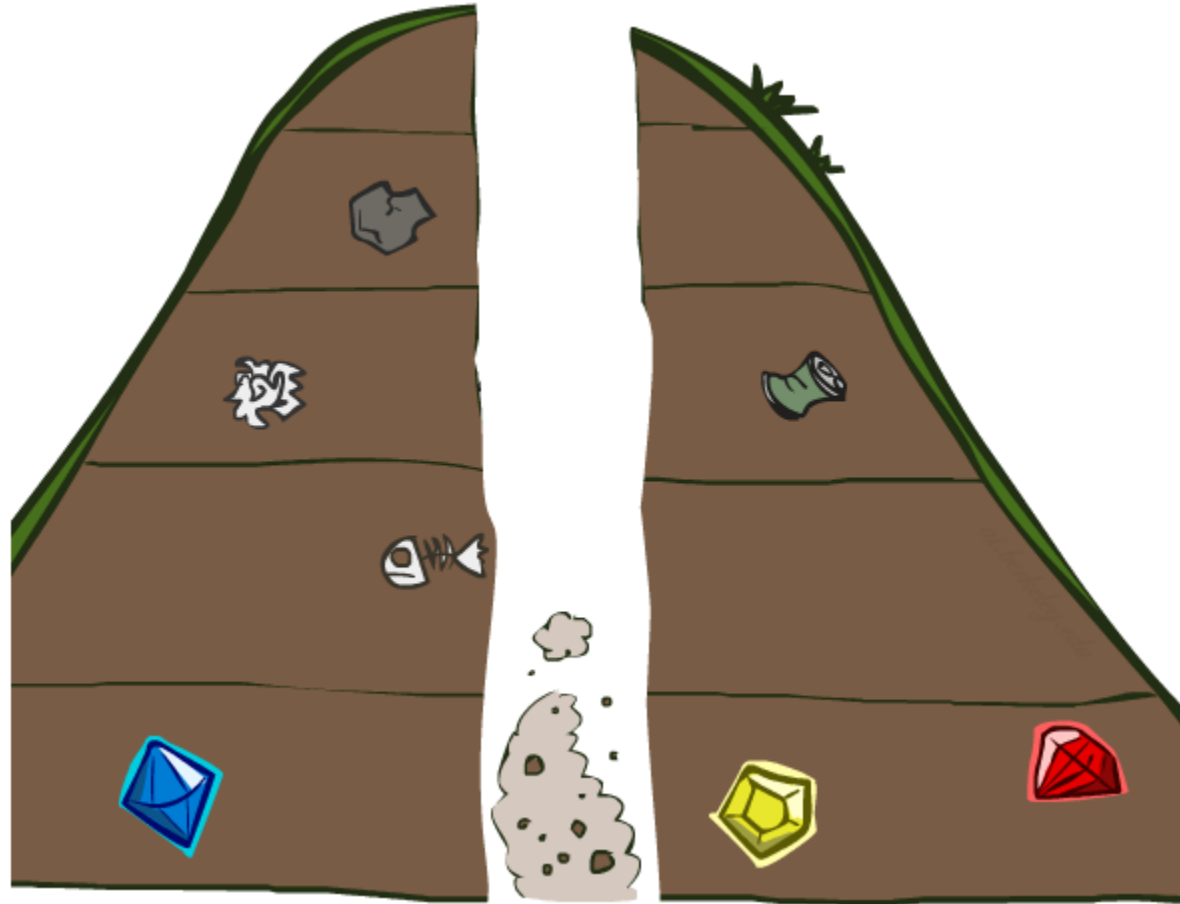
□ روش‌های جستجوی ناآگاهانه (کور)

- تنها از اطلاعات موجود در تعریف مسئله برای جستجو استفاده می‌کند.
- تنها می‌تواند حالت‌های هدف را از حالات غیر هدف تشخیص دهد.

□ انواع روش‌های جستجوی ناآگاهانه:

1. جستجوی عمقی (DFS)
2. جستجوی سطحی (BFS)
3. جستجوی عمقی افزایشی (ترکیب DFS و BFS) (IDS)
4. جستجوی هزینه یکنواخت (UCS)

ارزیابی الگوریتم‌های جستجو



ارزیابی الگوریتم‌های جستجو

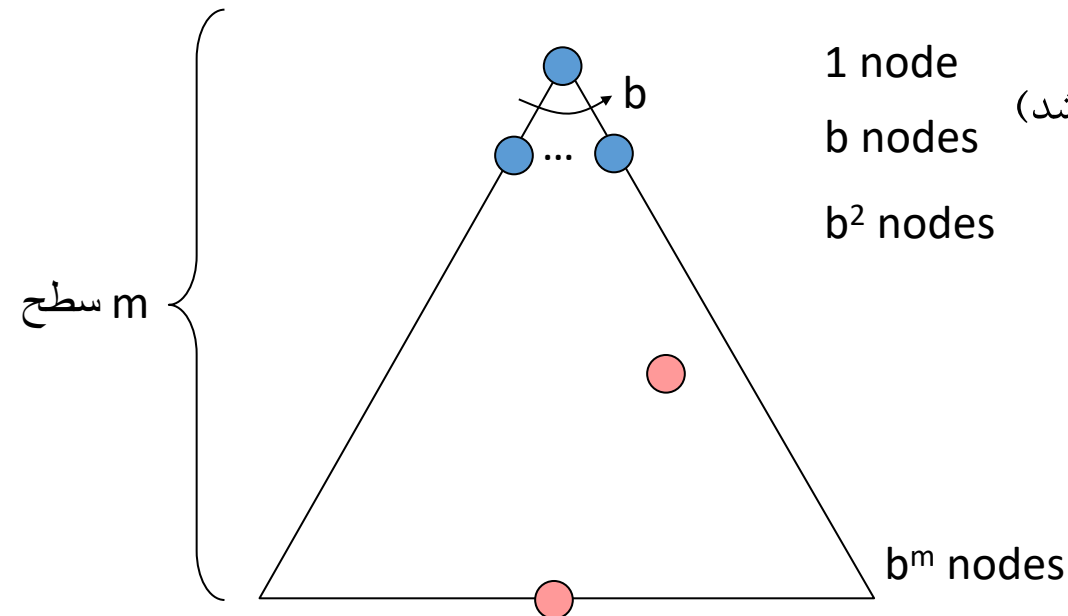
1. کامل بودن (Complete): آیا تضمین می‌کند که اگر راه حلی وجود داشته باشد، آن را پیدا کند؟
2. بهینه بودن (Optimal): آیا تضمین می‌کند که کم‌هزینه‌ترین راه (بهینه‌ترین راه) را پیدا کند؟
3. پیچیدگی زمانی (Time complexity): مقدار زمان مورد نیاز برای اجرای الگوریتم جستجو
4. پیچیدگی حافظه (Space complexity): مقدار حافظه مورد نیاز برای اجرای الگوریتم جستجو

□ پارامترهای مهم درخت جستجو:

- b ضریب انشعاب (تعداد فرزندان هر گره).
- m حداکثر عمق درخت (طولانی‌ترین مسیری که درخت جستجو می‌تواند داشته باشد)
- s نشان‌دهنده عمق اولین راه حل (کم‌عمق‌ترین راه حال) در درخت جستجو است.
- ✓ دقت شود راه حل‌ها در عمق‌های مختلف پراکنده هستند

□ تعداد کل گره‌های درخت جستجو:

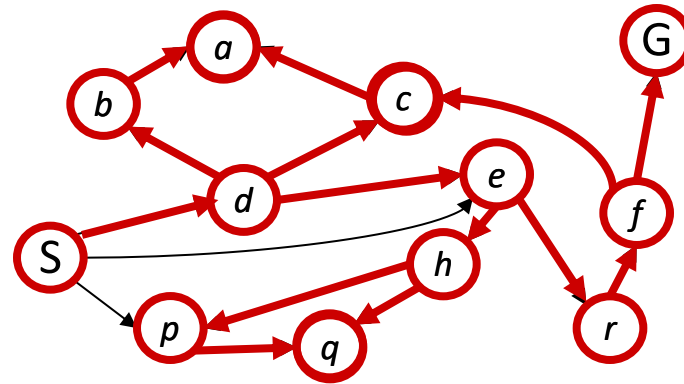
$$1 + b + b^2 + \dots + b^m = O(b^m)$$



۱. جستجوی عمقی (Depth-First Search - DFS)

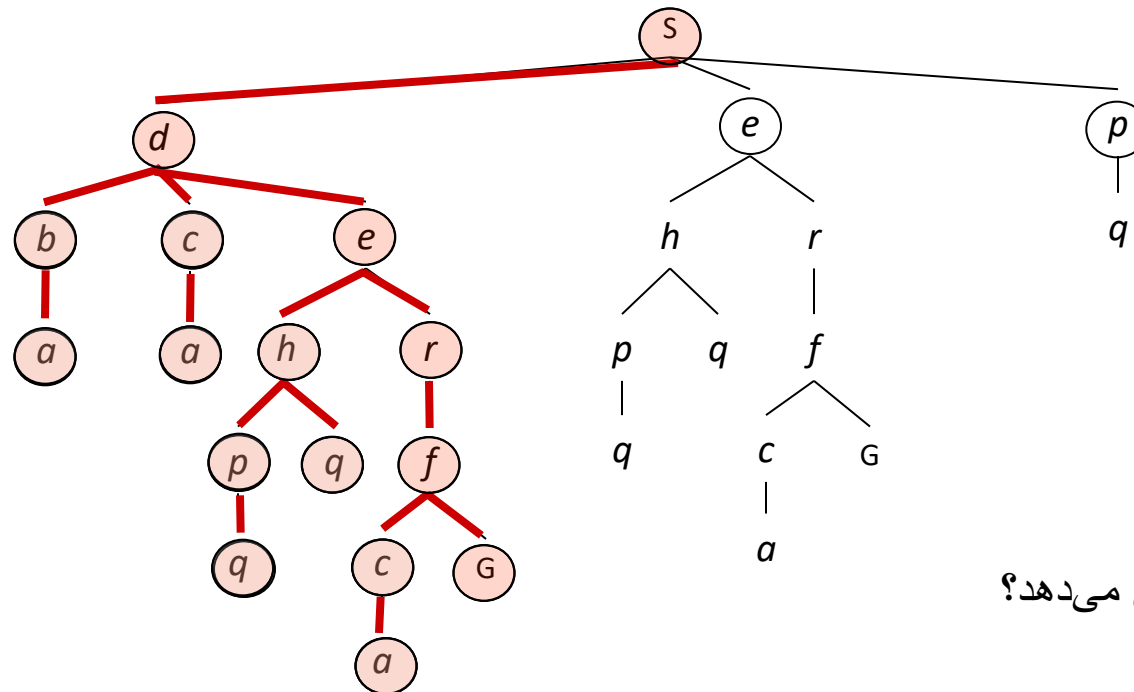


۱. جستجوی عمقی



استراتژی: ابتدا عمیق‌ترین گره را گسترش دهید.

پیاده سازی: حاشیه (Fringe) یک پشته LIFO است.



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