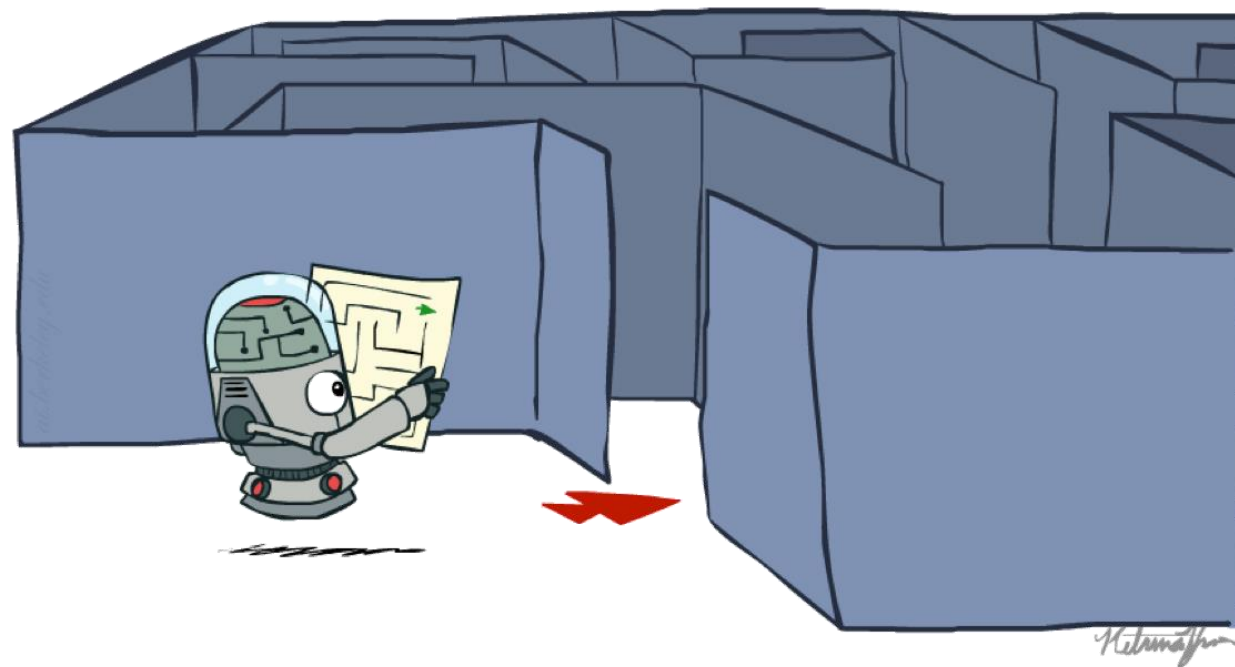


# حل مسائل توسط جستجو

سید ناصر رضوی [n.razavi@tabrizu.ac.ir](mailto:n.razavi@tabrizu.ac.ir)

۱۳۹۷



# فهرست مطالب

۳

□ عامل‌های مبتنی بر جستجو.

□ مسائل جستجو.

□ الگوریتم‌های جستجوی ناآگاهانه.

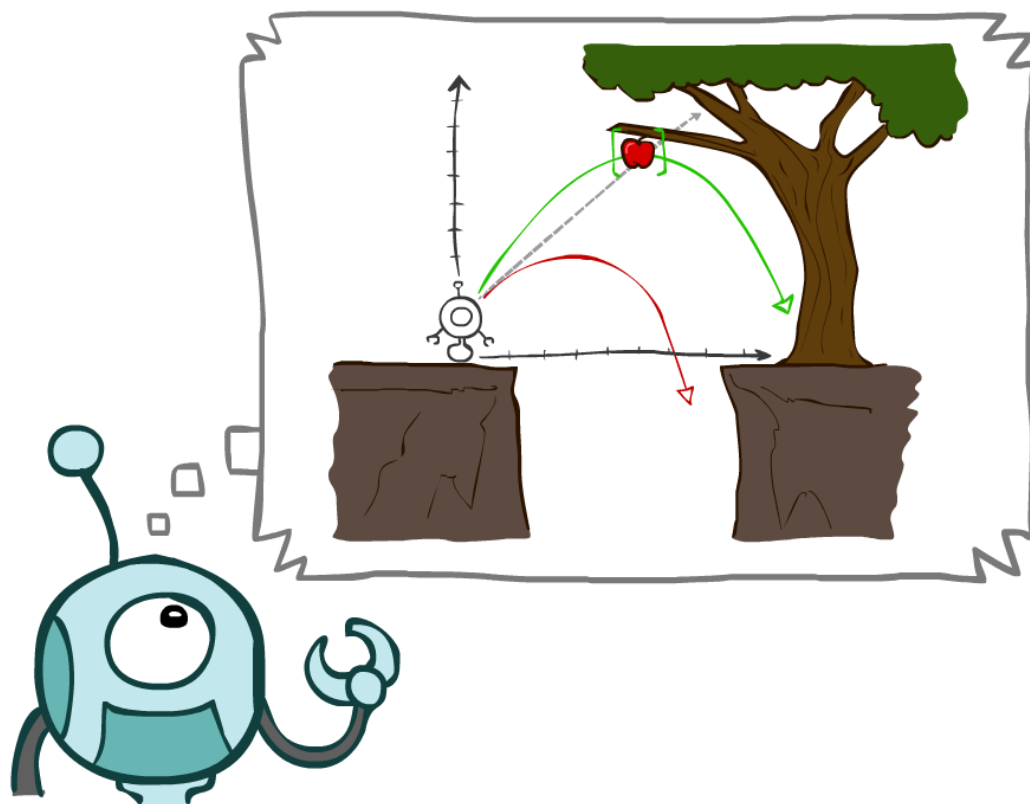
□ جستجوی سطحی

□ جستجوی هزینه یکنواخت

□ جستجوی عمقی

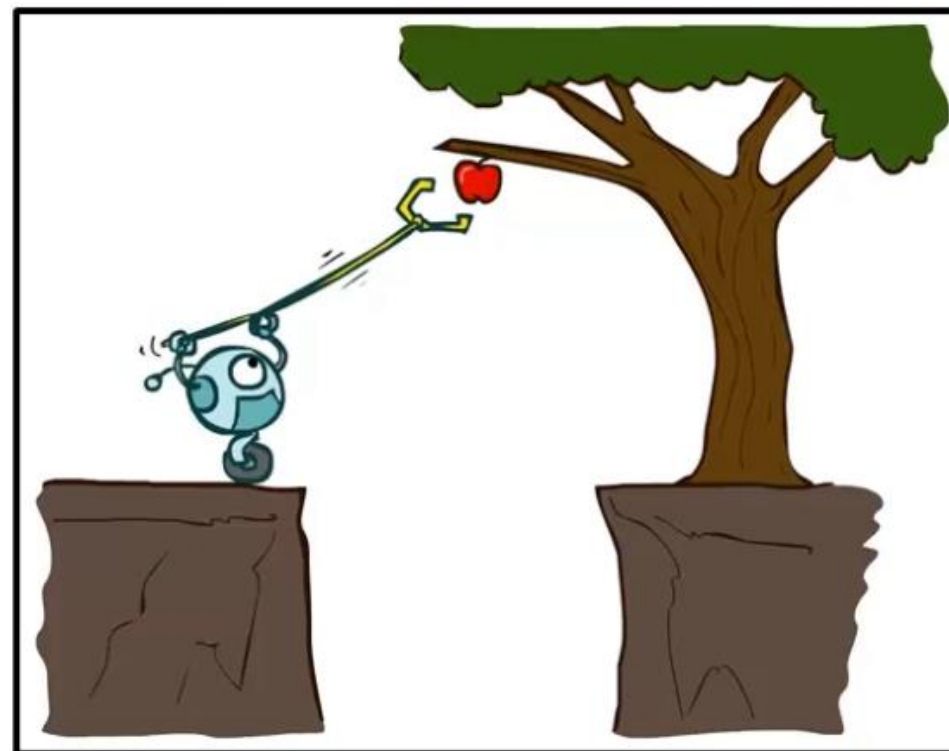
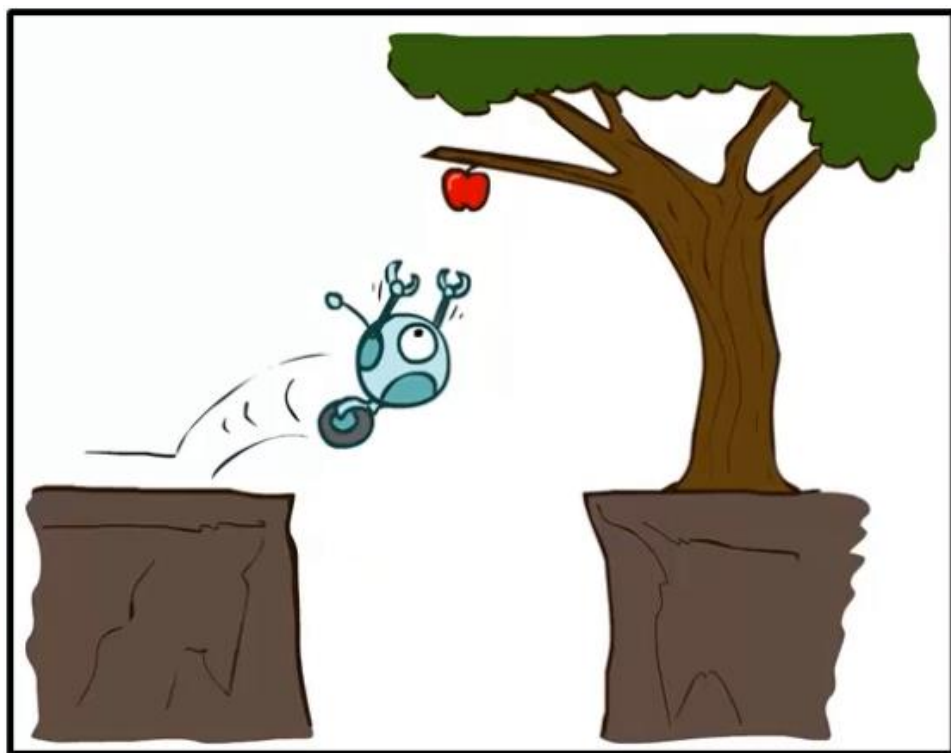
□ جستجوی عمقی محدود

□ جستجوی عمیق کننده تکراری



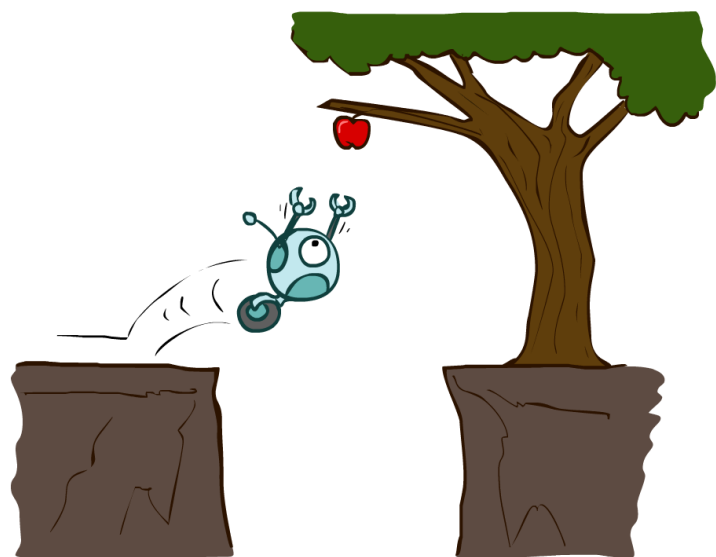
# عامل مبتنی بر جستجو

۴



# عامل واکنشی

۵



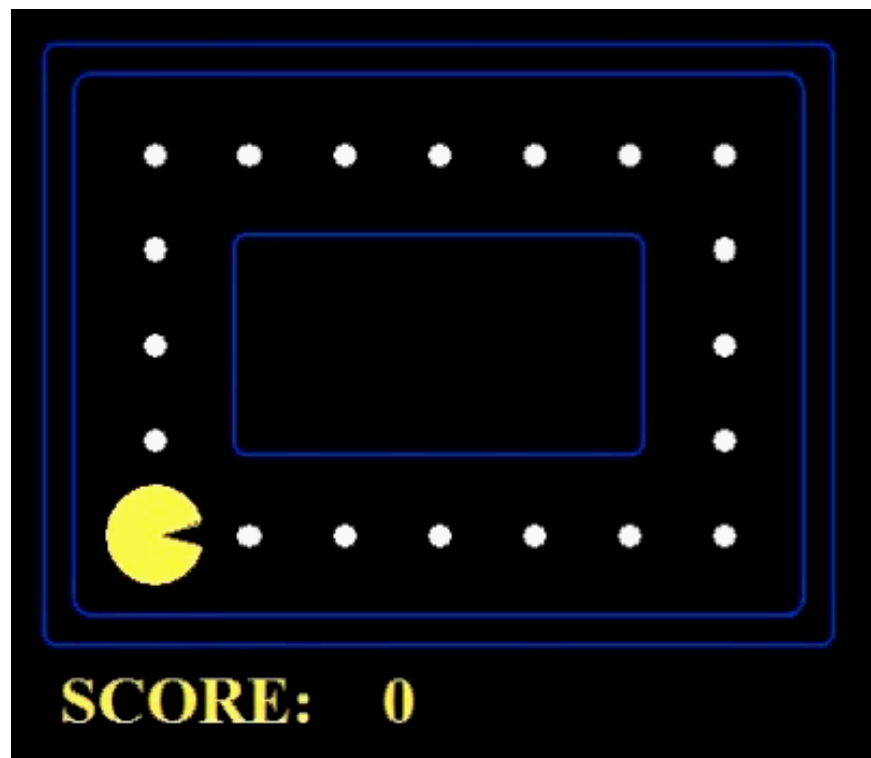
## □ عامل واکنشی.

- انتخاب عمل تنها بر اساس **درک فعلی** از محیط
- می‌تواند **مدلی** از حالت فعلی محیط داشته باشد
- تنها حالت فعلی محیط برایش اهمیت دارد
- عواقب عمل انتخاب شده را در نظر نمی‌گیرد
- شامل مجموعه‌ای از **قوانین شرط-عمل**

□ **س.** آیا یک عامل واکنشی می‌تواند منطقی عمل کند؟

# عامل واکنشی: موفقیت

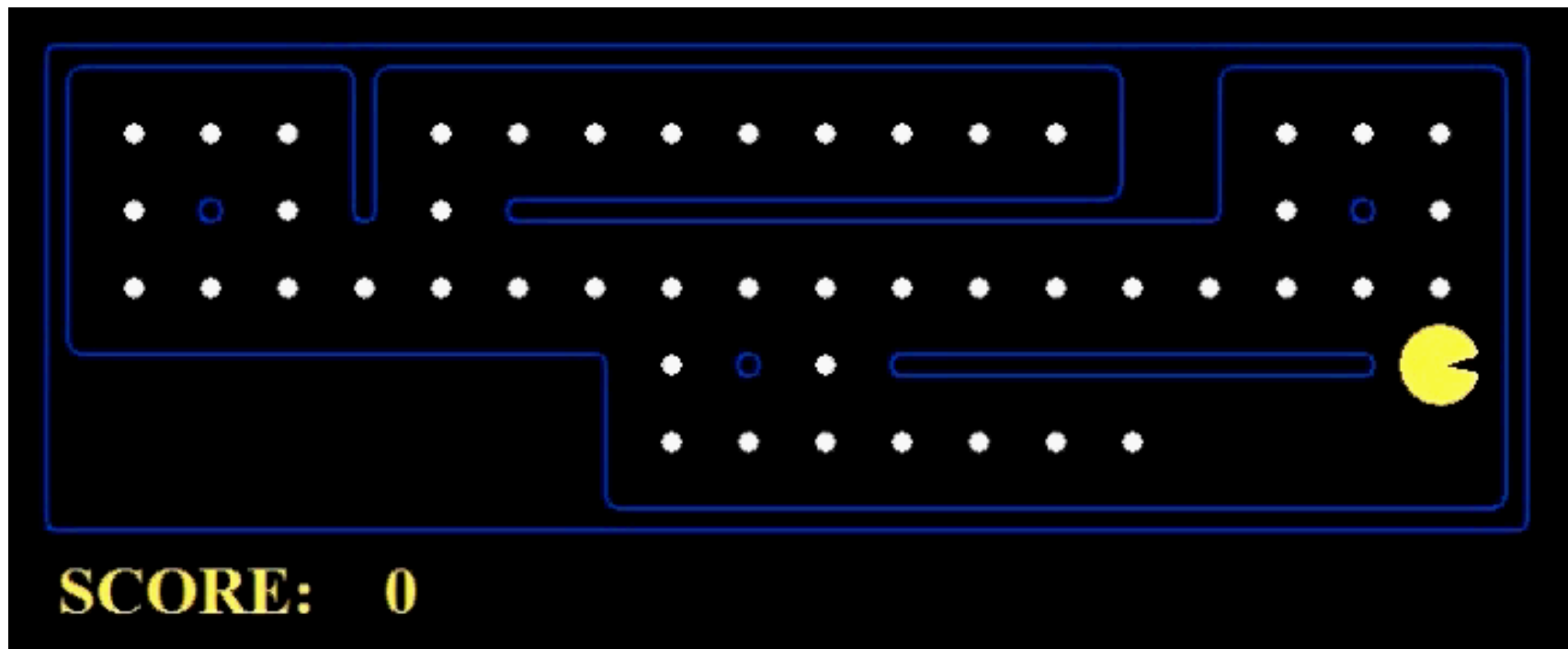
۶



اگر در خانه مجاور غذا وجود دارد، آنگاه آن را بخور

# عامل واکنشی: شکست

۷

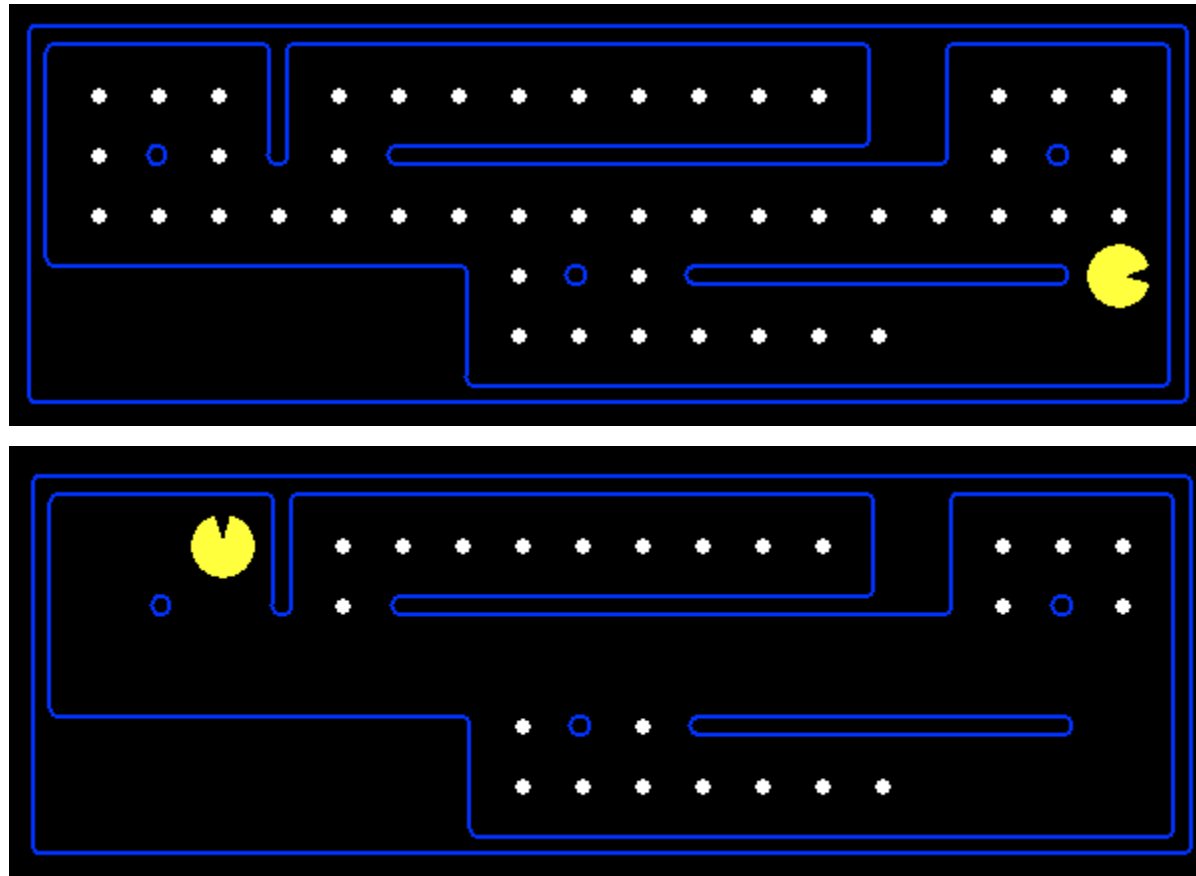


اگر در خانه مجاور غذا وجود دارد، آنگاه آن را بخور

# عامل واکنشی: شکست

۸

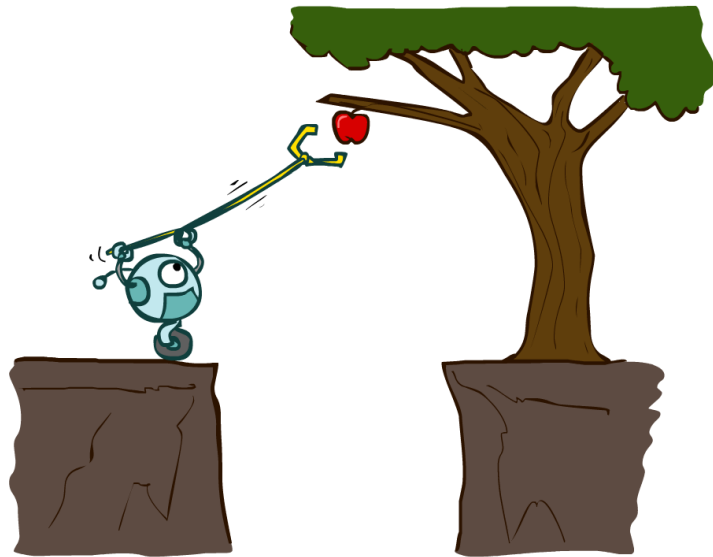
اگر در خانه مجاور غذا وجود دارد، آنگاه آن را بفور





# عامل هدف‌گرا (برنامه‌ریزی کننده)

۹



## □ عامل هدف‌گرا.

□ دارای هدف

□ می‌پرسد «چه می‌شود اگر این عمل را انجام دهم»

□ تصمیم‌گیری بر اساس عواقب فرضی اعمال [شبیه‌سازی]

□ دارای یک مدل از حالت فعلی محیط و دانش درباره:

■ چگونگی تغییر محیط

■ تاثیر اعمال خود بر محیط

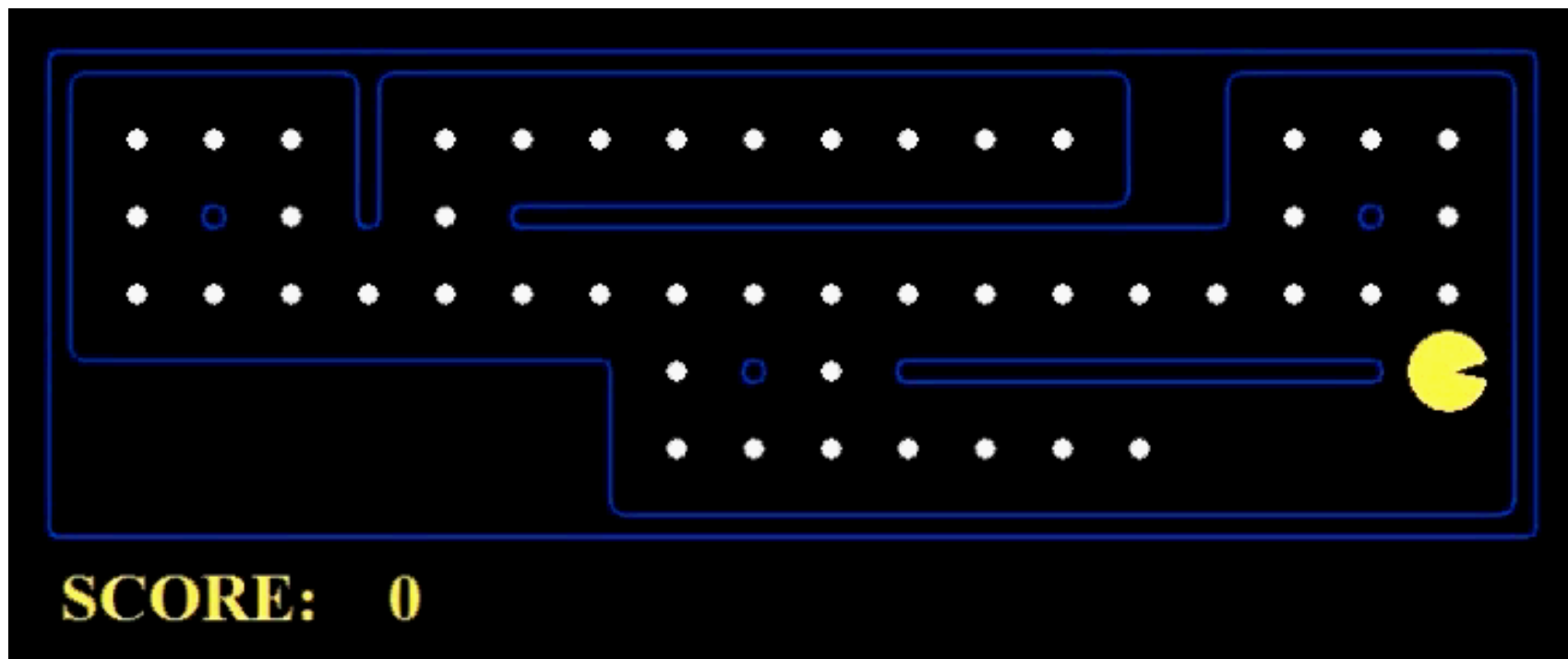
## □ برنامه‌ریزی کامل و برنامه‌ریزی بهینه.

□ برنامه‌ریزی کامل: در صورت وجود راه‌حل حتماً آن را پیدا می‌کند.

□ برنامه‌ریزی بهینه: همواره بهترین راه‌حل ممکن را پیدا می‌کند.

# عامل هدف‌گرا (برنامه‌ریزی کننده)

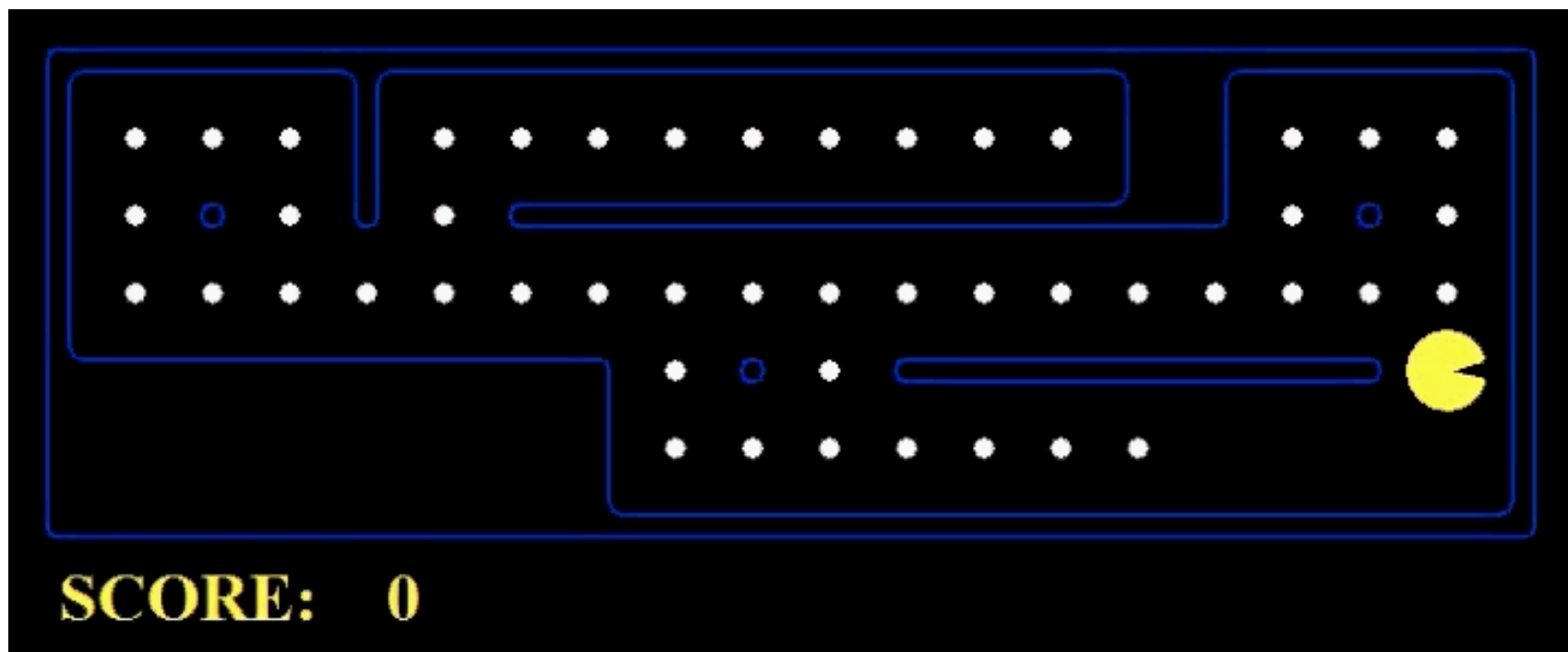
۱۰



یافتن یک مسیر تا یک غذا، اجرای مسیر، تکرار این مراحل تا خوردن همه غذاها

# عامل هدف‌گرا (برنامه‌ریزی کننده)

۱۱



هدف. خوردن تمام غذاها با حداقل تعداد عملیات ممکن

# پرسش کلاسی: عامل واکنشی یا برنامه‌ریز

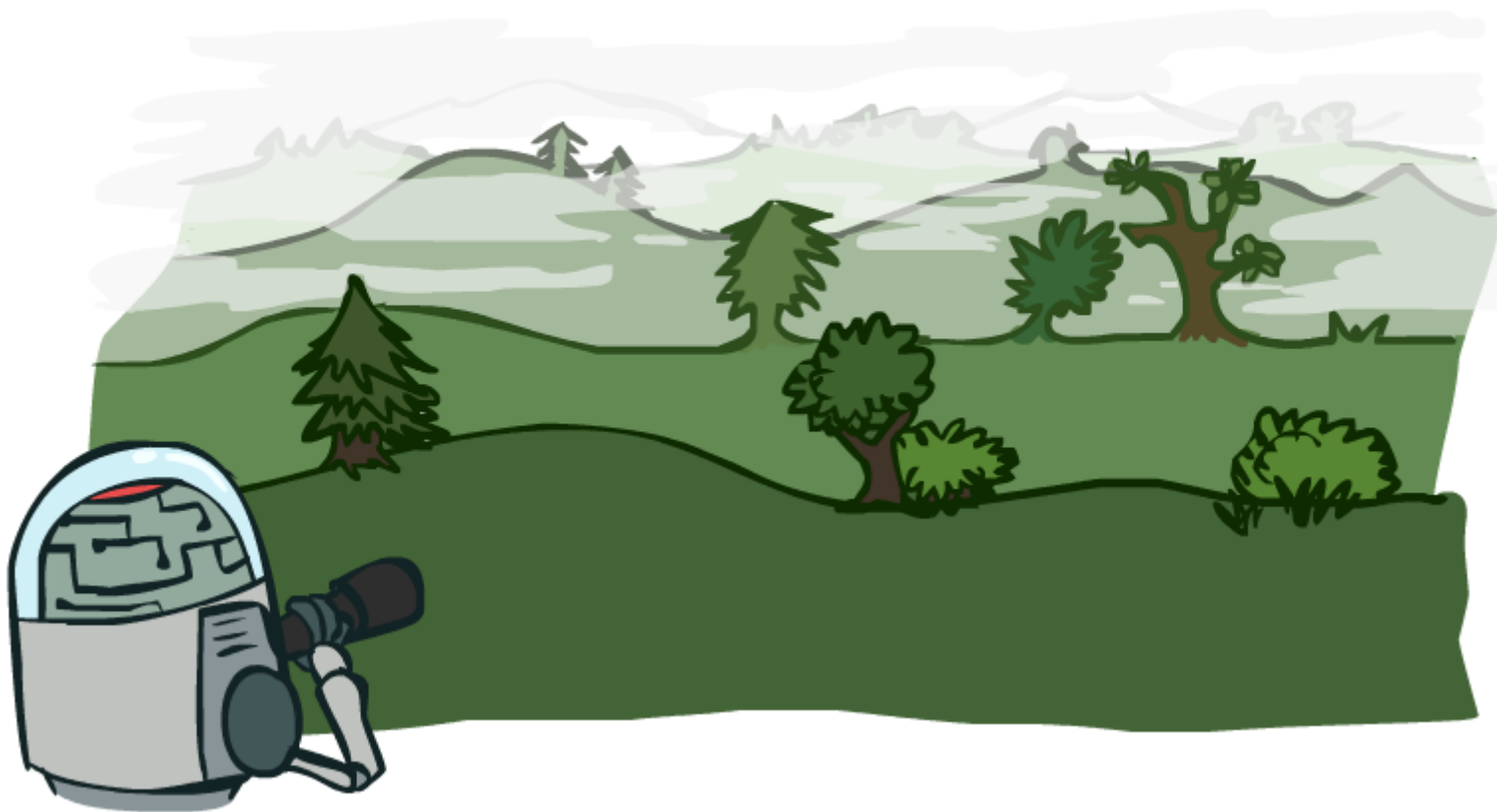
۱۲

□ س. در هریک از موارد زیر تعیین کنید عامل ارائه شده از کدام نوع است: واکنشی یا برنامه‌ریزی کننده؟

- i. یک عامل پکمن که همیشه به سمت نزدیک‌ترین غذا حرکت می‌کند.
- ii. یک عامل پکمن که همیشه به سمت نزدیک‌ترین غذا حرکت می‌کند، مگر این که در مسیر حرکتش یک روح با فاصله‌ای کمتر از سه خانه وجود داشته باشد.
- iii. یک عامل مسیریاب که تمام مسیرها تا مقصد را در نظر می‌گیرد و سپس کوتاه‌ترین مسیر را انتخاب می‌کند.

# مسائل جستجو

۱۳



# مسایل جستجو

۱۴

□ یک مسئله جستجو شامل موارد زیر است:

□ یک فضای حالت

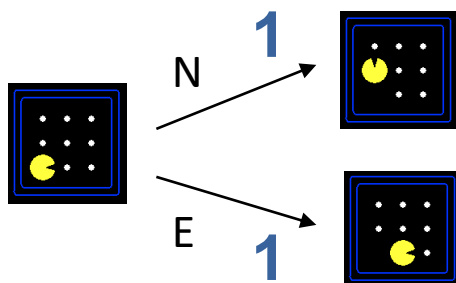
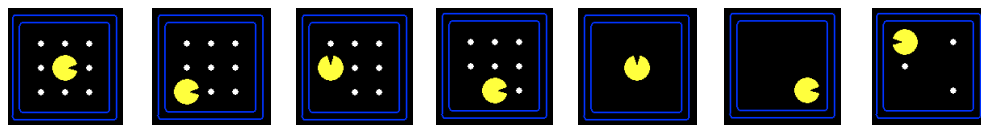
□ یک تابع جانشین:

■ بیانگر عملیات قابل انجام در هر حالت و هزینه و نتیجه هر عمل

■ بیانگر چگونگی تغییر محیط در پاسخ به اعمال عامل

□ یک حالت شروع و یک حالت هدف

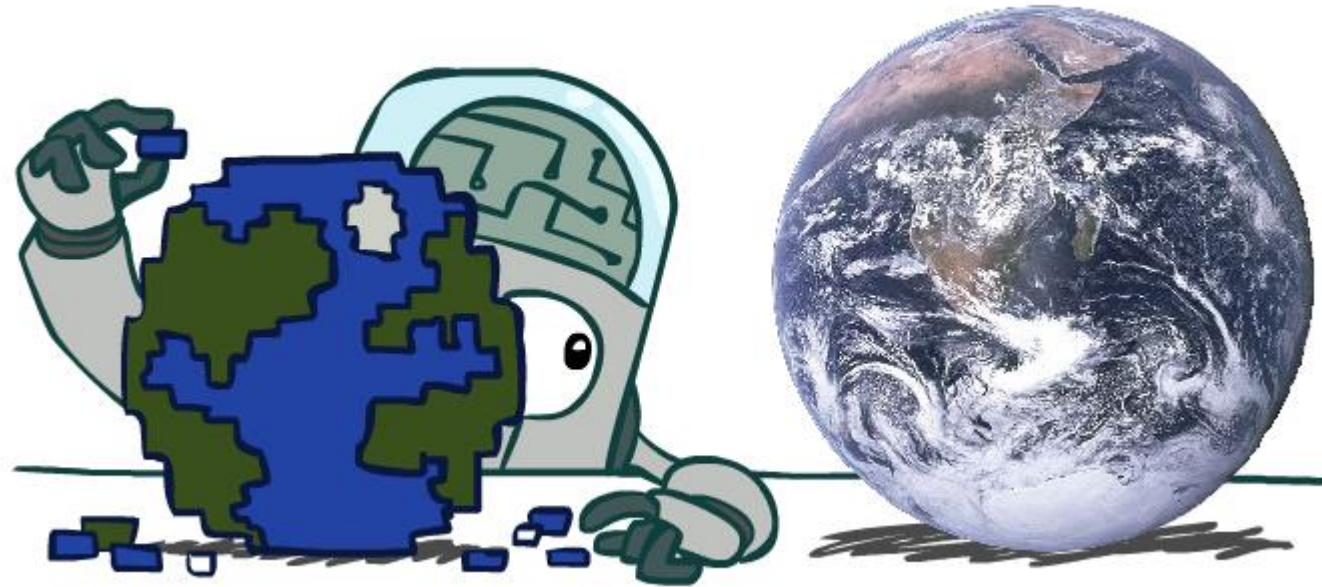
□ راه حل. دنباله‌ای از عملیات که حالت شروع را به یک حالت هدف تبدیل می‌کند.



# مدل سازی (انتزاع)

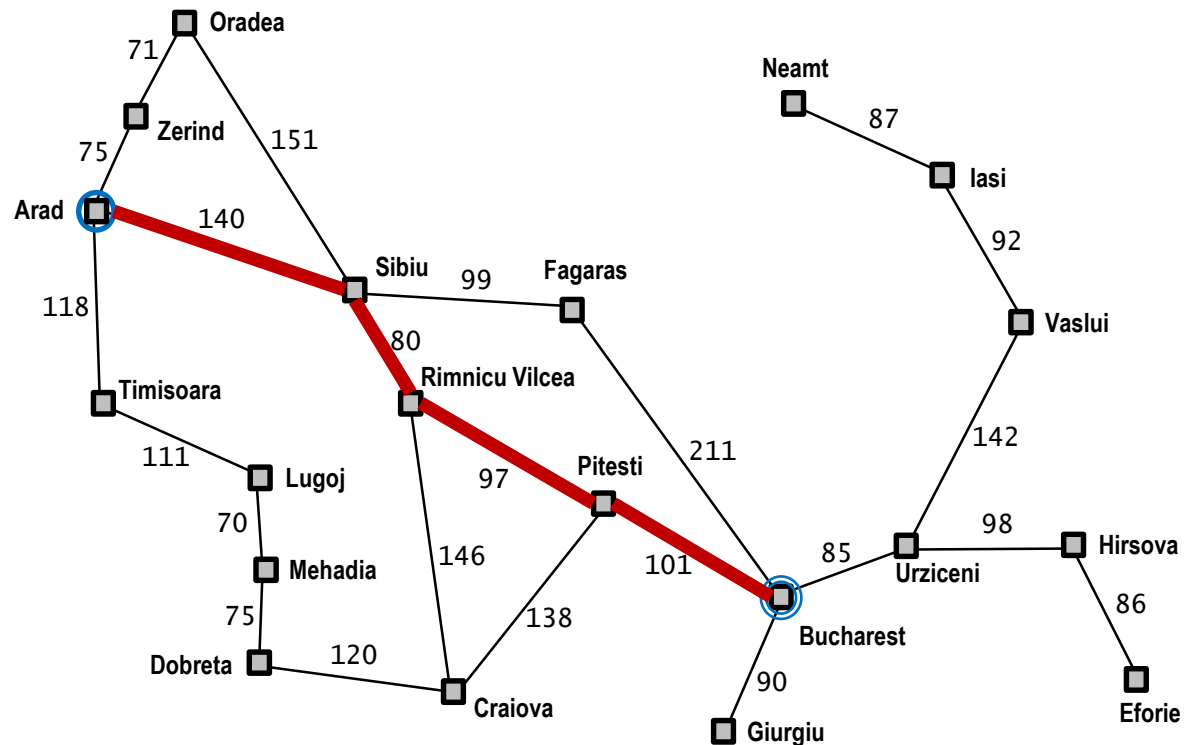
۱۵

□ **مدل سازی.** مسائل جستجو تنها یک **مدل** از مسئله واقعی هستند.  
□ حالت ها و عملیات انتزاعی هستند!



# مثال: مسیریابی در رومانی

۱۶



□ فضای حالت

□ شهرها

□ تابع جانشین

□ جاده‌ها، رفتن به یک شهر همسایه

□ هزینه = فاصله

□ حالت شروع

□ آراد

□ آزمایش هدف

□ بودن در بخارست

راه‌حل. یک مسیر از آراد به بخارست.



# حالت مسئله

۱۷

□ س. حالت مسئله شامل چه اطلاعاتی است؟

□ ج. حالت مسئله تنها شامل جزییات لازم برای جستجو است. [انتزاع]

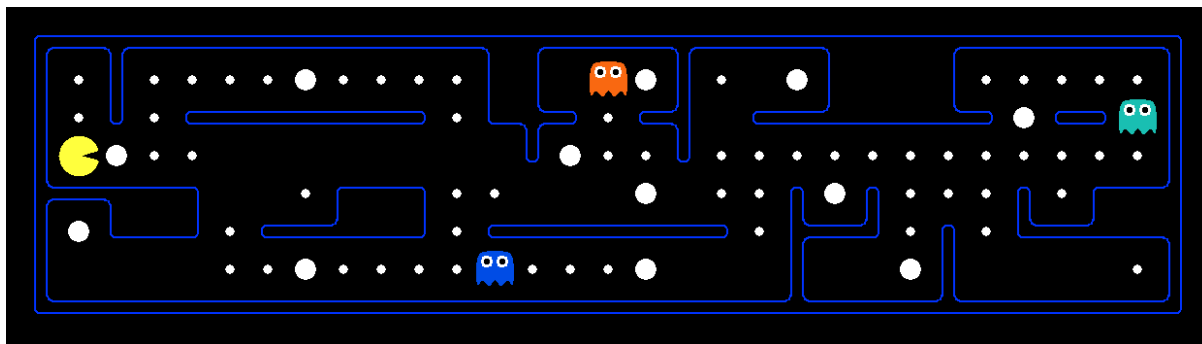
□ مسئله: مسیریابی

□ حالت‌ها: زوج مرتب  $(x,y)$  بیانگر مکان عامل

□ عملیات: شمال، جنوب، شرق، غرب

□ حالت بعدی: به روز رسانی مکان عامل

□ آزمایش هدف: آیا  $(x,y) = (x_G, y_G)$



□ س. حالت مسئله شامل چه اطلاعاتی است؟

□ ج. حالت مسئله تنها شامل جزییات لازم برای جستجو است. [انتزاع]

□ مسئله: خوردن تمام غذاها

□ حالت‌ها: مکان عامل + یک ماتریس بولی بیانگر این که در کدام خانه‌ها غذا وجود دارد

□ عملیات: شمال، جنوب، شرق، غرب

□ حالت بعدی: به روز رسانی مکان عامل و احتمالاً یکی از خانه‌های ماتریس بولی

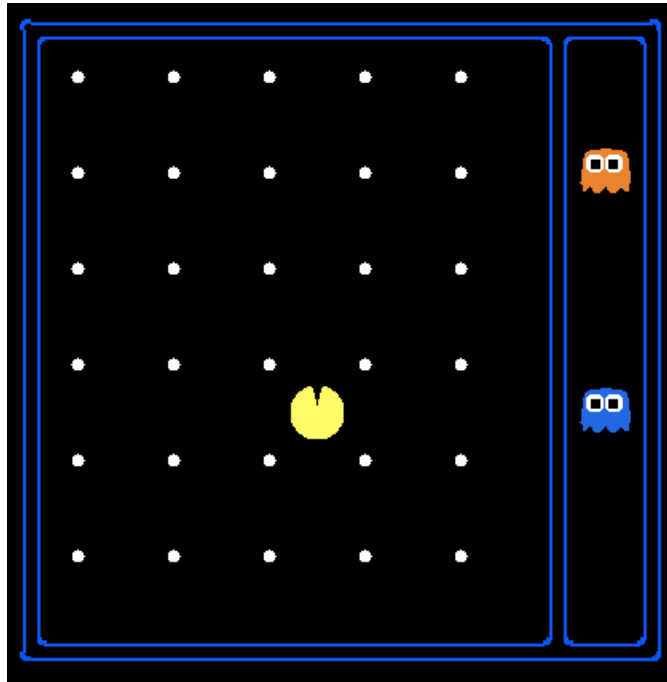
□ آزمایش هدف: هیچ غذایی باقی نمانده باشد [همه عناصر ماتریس بولی برابر با false]

# اندازه فضای حالت

۱۹

□ اندازه فضای حالت. تعداد حالت‌های قابل دسترس از حالت شروع.

$$۱۲۰ \times ۴ \times ۱۲^۲ \times ۳۰$$



□ فضای حالت:

- مکان‌های ممکن برای عامل: ۱۲۰
- جهت‌های ممکن برای عامل: ۴
- حالت‌های ممکن برای ارواح: ۱۲
- تعداد غذاها: ۳۰

# اندازه فضای حالت

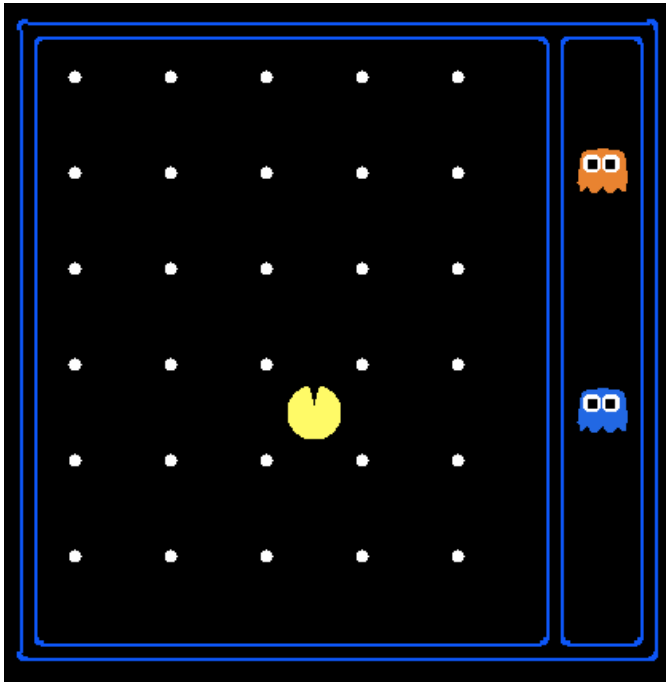
۲۰

□ اندازه فضای حالت. تعداد حالت‌های قابل دسترس از حالت شروع.

□ اندازه فضای جستجو:

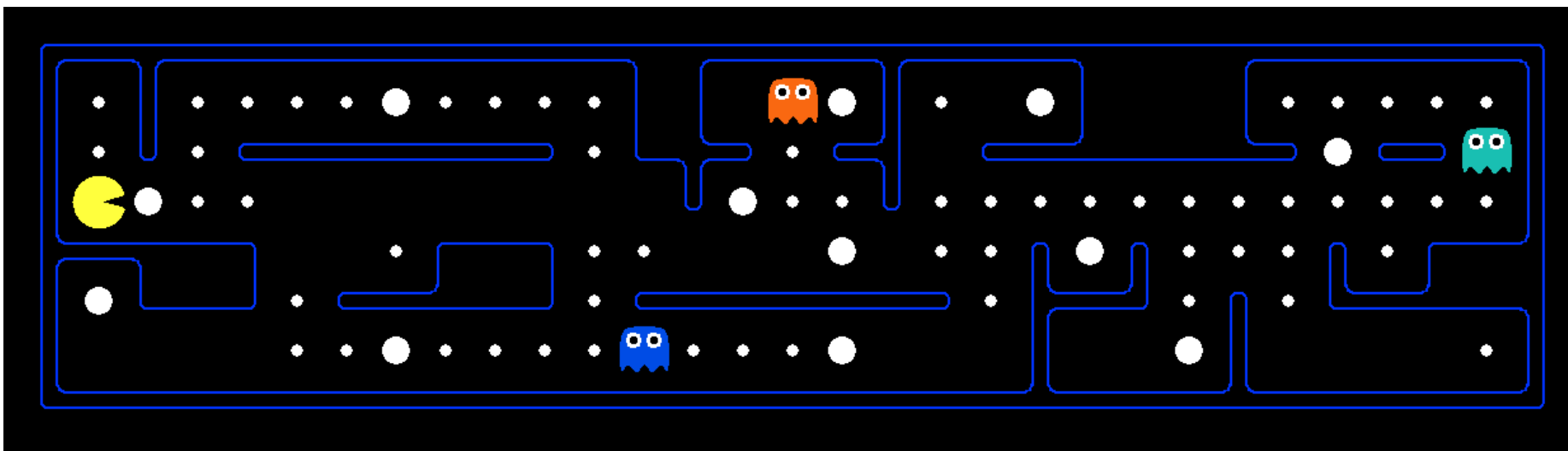
□ مسئله مسیریابی: ۱۲۰

□ مسئله خوردن تمام غذاها:  $120 \times 2^{30}$



# پرسش کلاسی: عبور امن

۲۱



❑ مسئله. خوردن تمام غذاها و در ترس نگه داشتن ارواح به طور همزمان.

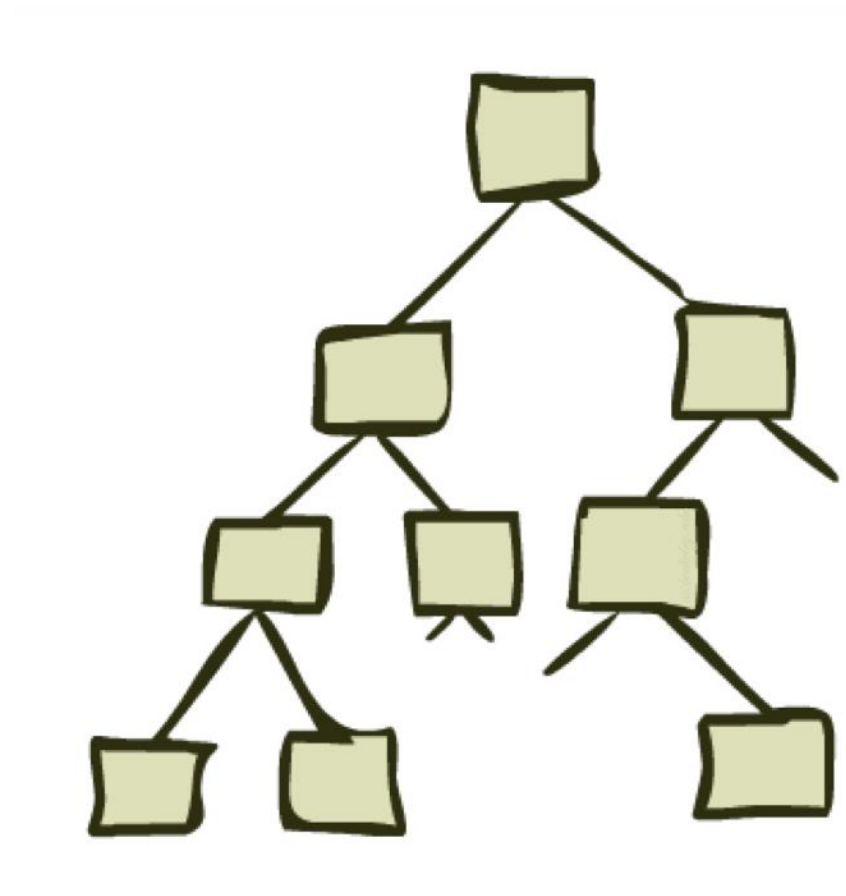
❑ س. چه اطلاعاتی باید در حالت‌ها ذخیره شوند؟

❑ مکان غذاها، مکان کیسول‌های انرژی

❑ مدت زمان باقیمانده از زمان ترس هر یک از ارواح!

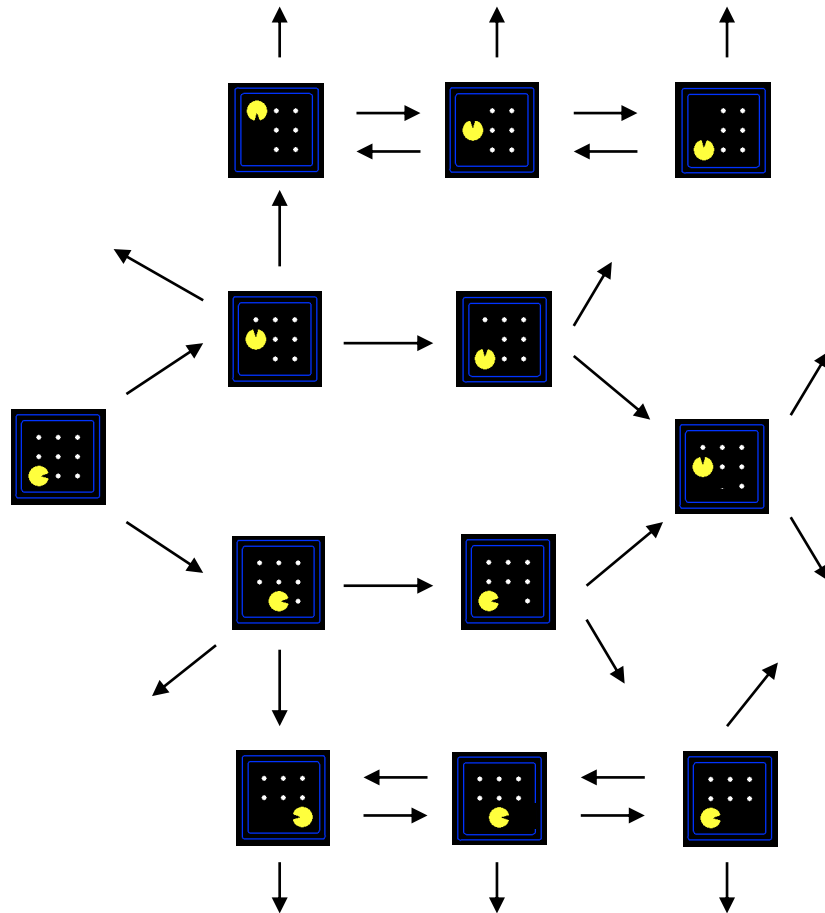
# گراف فضای حالت و درخت جستجو

۲۲



# گراف فضای حالت

۲۳



□ گراف فضای حالت. یک مدل ریاضی از مسئله جستجو

□ رئوس: حالت‌های محیط (انتزاعی)

□ یال‌ها: عملیات ممکن در هر رأس

□ تابع حالت بعدی: رئوس همسایه

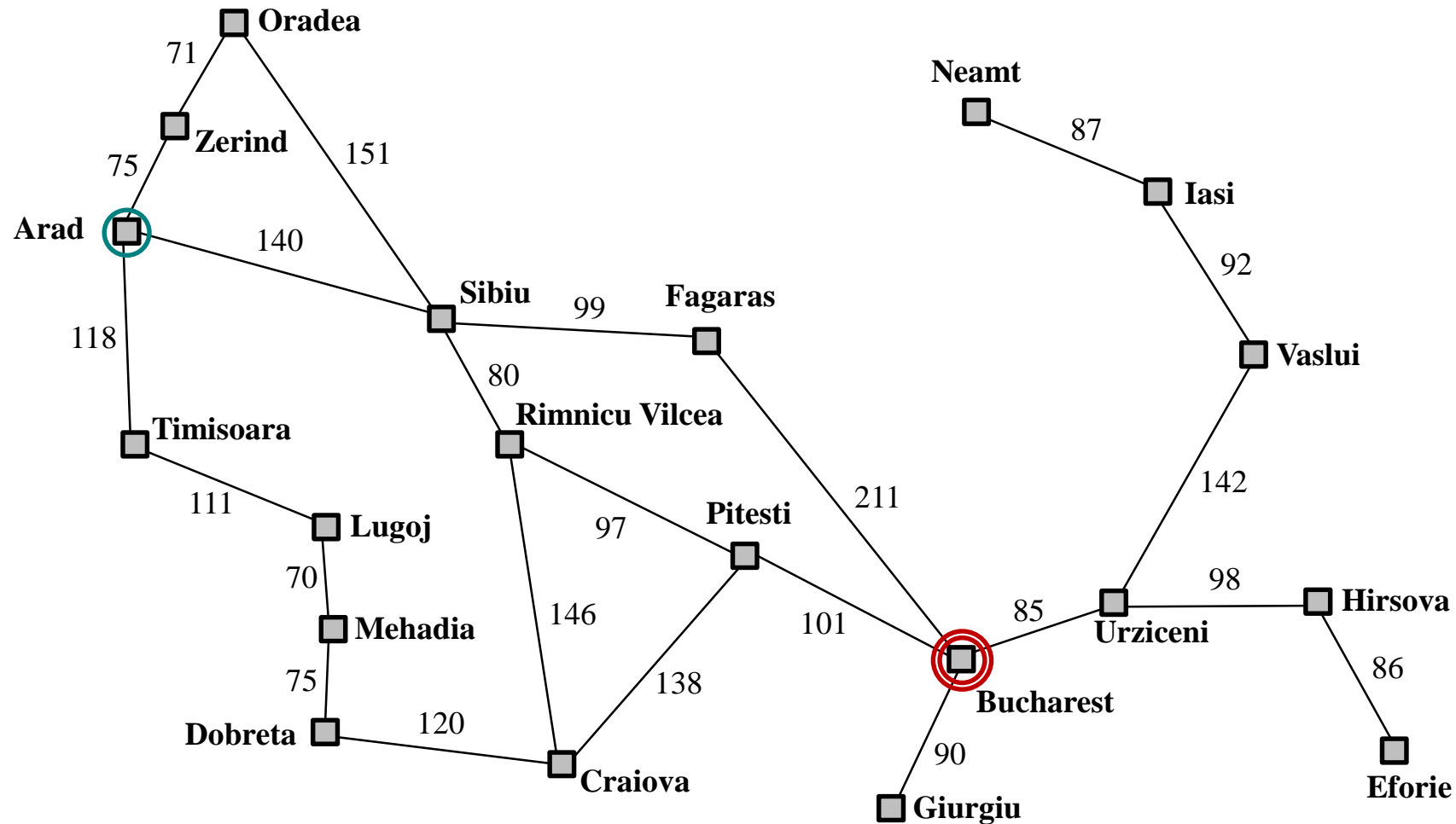
□ حالت شروع: یکی از رئوس

□ آزمون هدف: مجموعه‌ای از یک یا چند رأس

□ توجه. در گراف فضای حالت، رأس تکراری وجود ندارد!

# گراف فضای حالت: مسیریابی در رومانی

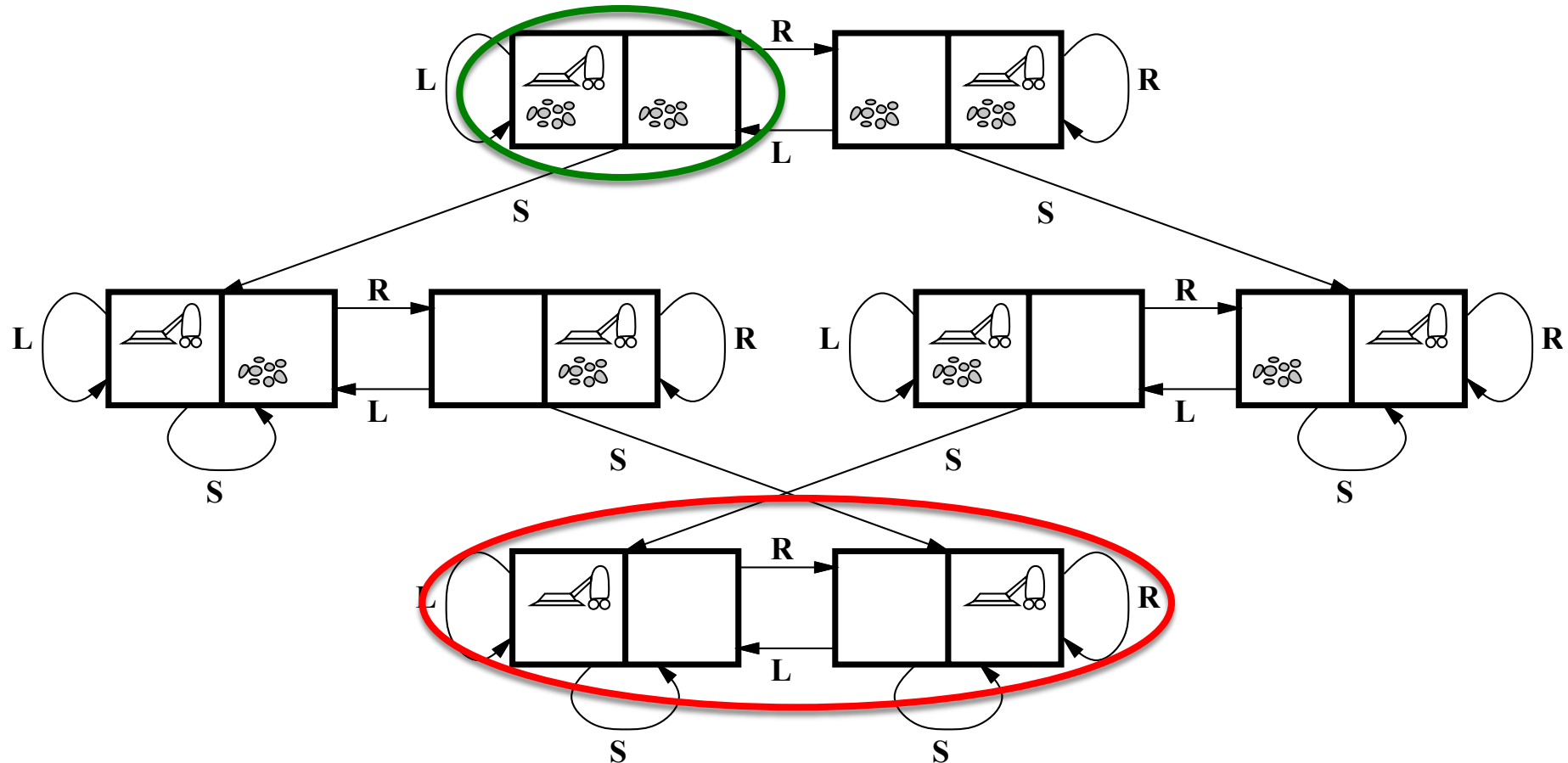
۲۴





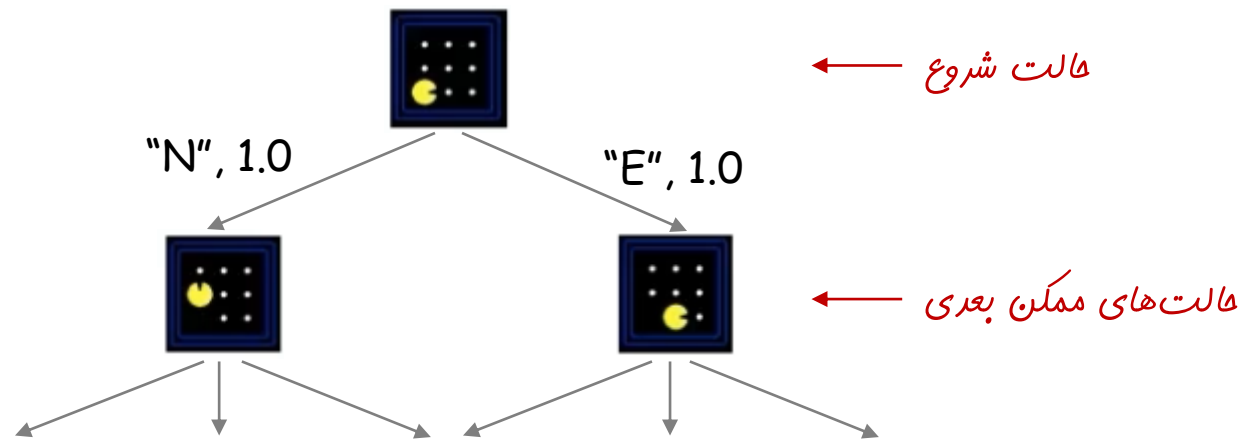
# گراف فضای حالت: دنیای مکش

۲۵



# درخت جستجو

۲۶



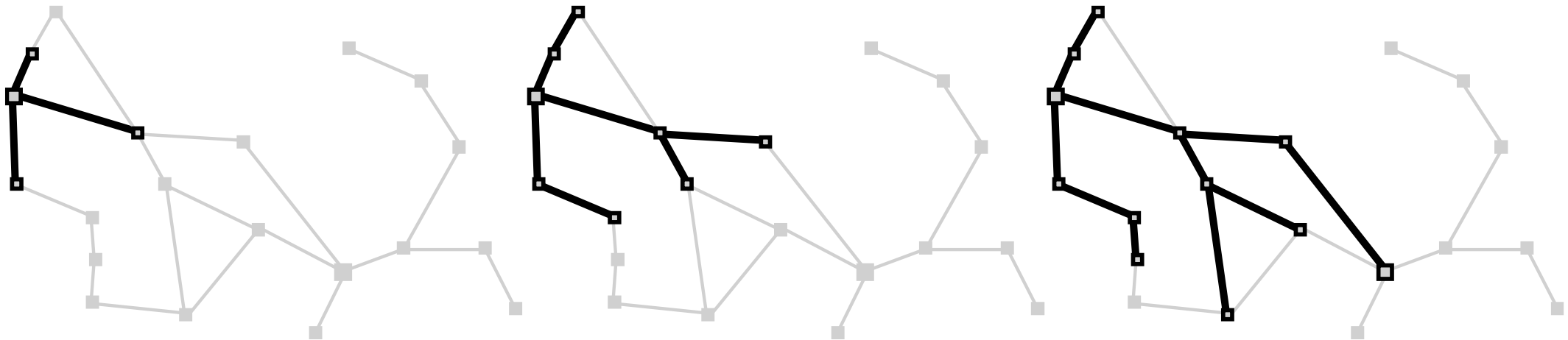
□ درخت جستجو.

- ریشه برابر با حالت شروع مسئله است.
- فرزندان یک گره متناظر با حالت‌های بعدی آن گره هستند.
- گره‌ها شامل حالت‌های مسئله هستند و مسیر رسیدن به آن حالت‌ها را نشان می‌دهند.
- در اغلب مسائل، ساختن تمامی درخت غیرممکن است!!!

# گراف فضای حالت و درخت جستجو

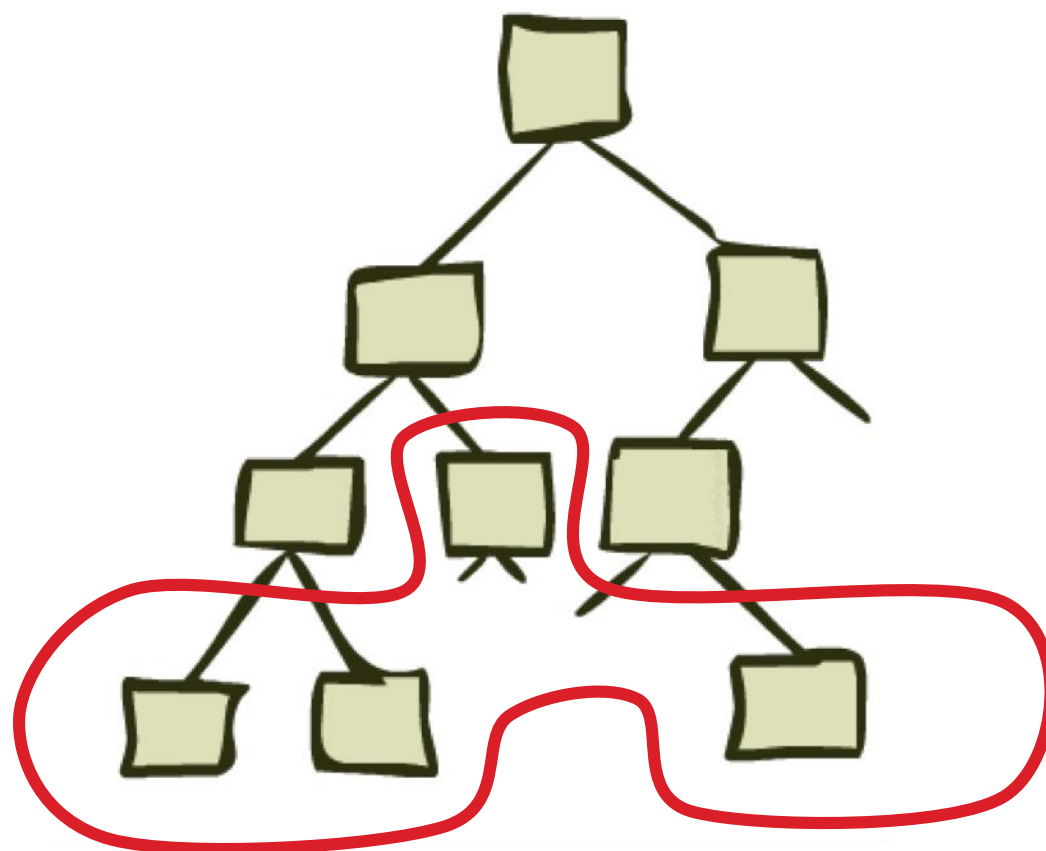
۲۷

□ توجه. گراف فضای حالت و درخت جستجو دو ساختار کاملاً متفاوت هستند و نباید با یکدیگر اشتباه گرفته شوند.



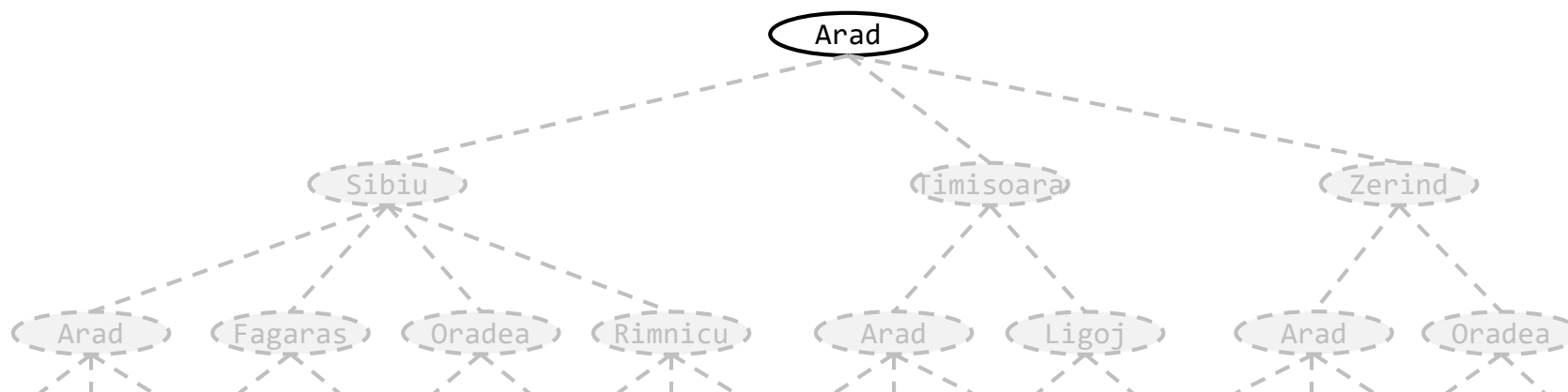
2人





# جستجوی فضای حالت با استفاده از درخت جستجو

۳۰



□ جستجو.

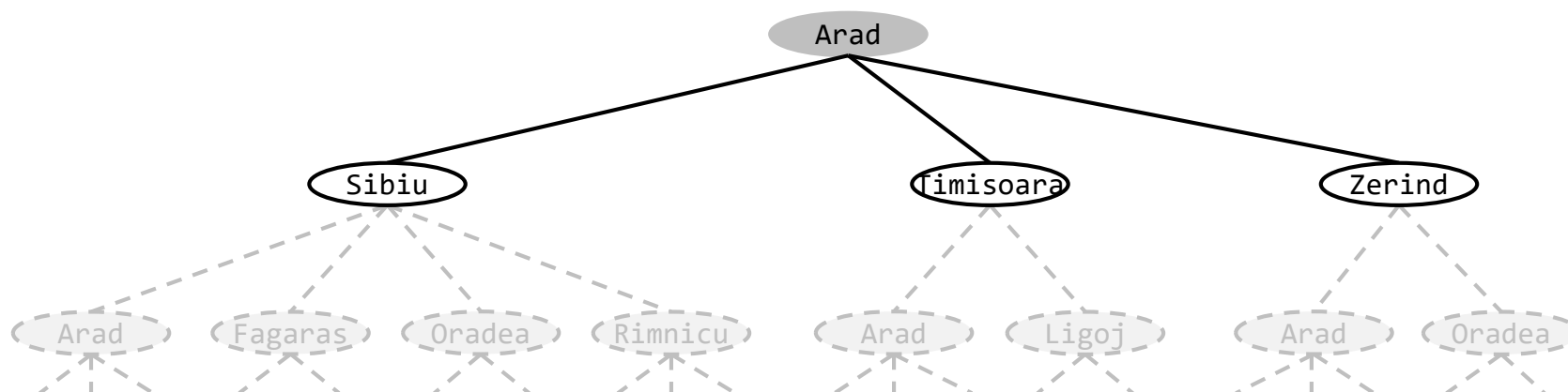
□ انتخاب یک گره نامزد

□ آزمایش هدف

□ گسترش گره انتخاب شده

# جستجوی فضای حالت با استفاده از درخت جستجو

۳۱



□ جستجو.

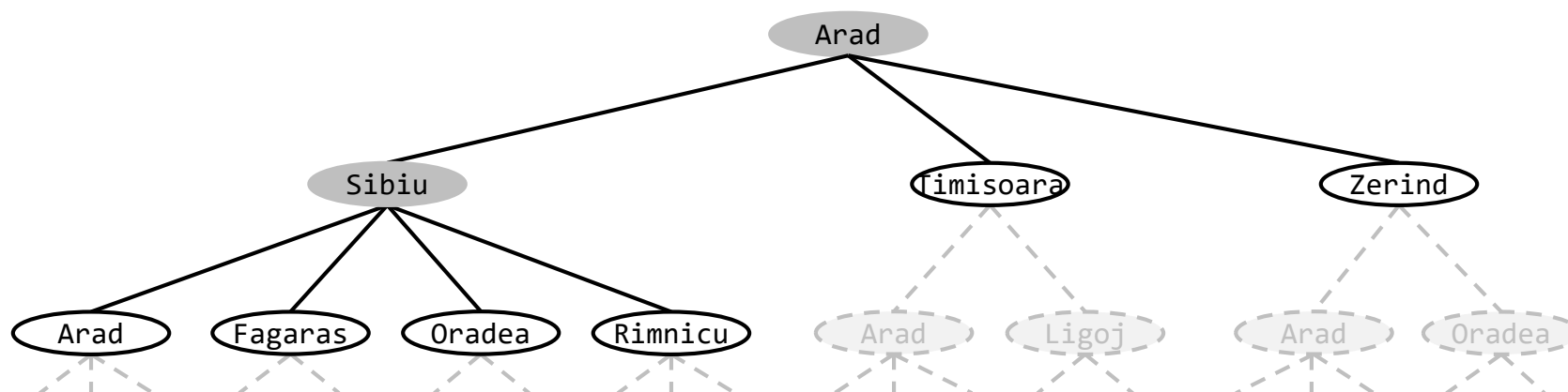
□ انتخاب یک گره نامزد

□ آزمایش هدف

□ گسترش گره انتخاب شده

# جستجوی فضای حالت با استفاده از درخت جستجو

۳۲



جستجو. □

□ انتخاب یک گره نامزد

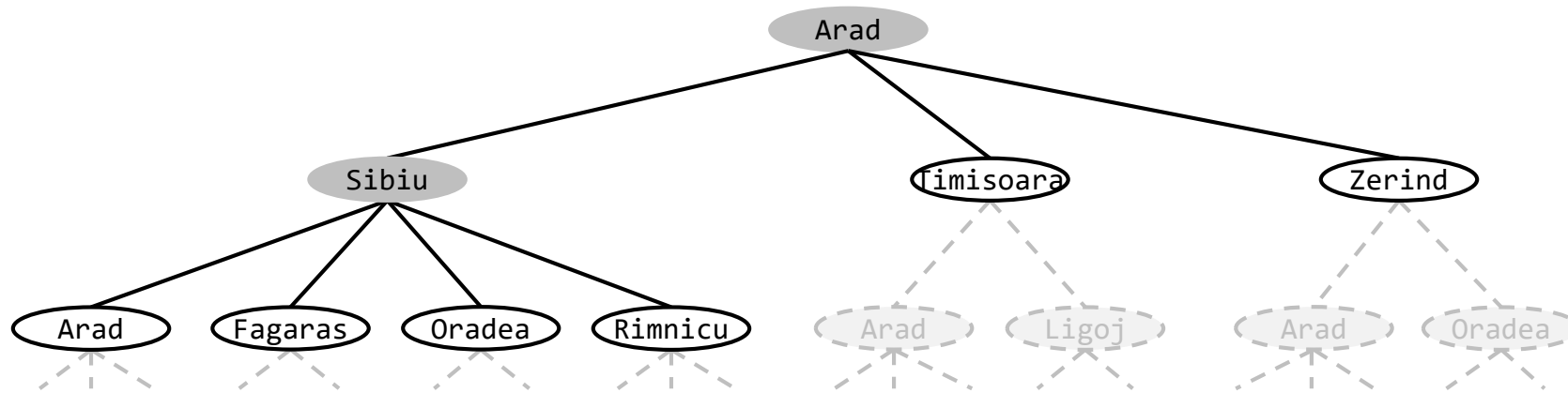
□ آزمایش هدف

□ گسترش گره انتخاب شده



# جستجوی فضای حالت با استفاده از درخت جستجو

۳۳



```
function TREE-SEARCH(problem, strategy) returns a solution, or failure
  initialize the search tree using the initial state of the 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
```

□ جستجو.

□ انتخاب یک گره نامزد

□ آزمایش هدف

□ گسترش گره انتخاب شده

# الگوریتم عمومی جستجوی درختی

۳۴

```
function TREE-SEARCH(problem, strategy) returns a solution, or failure
  initialize the search tree using the initial state of the 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
```

□ ایده‌های مهم.

□ استراتژی گسترش دادن

□ ساختمان داده fringe

□ گسترش دادن گره‌ها

هر بار کدام گره درخت باید گسترش داده شود؟

# الگوریتم عمومی جستجوی درختی

۳۵

```
function TREE-SEARCH(problem) returns a solution, or failure
  initialize the frontier using the initial state of problem
  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
    expand the chosen node, adding the resulting nodes to the frontier
  end
```

□ ایده‌های مهم.

□ استراتژی گسترش دادن

□ ساختمان داده fringe

□ گسترش دادن گره‌ها

هر بار کدام گره درخت باید گسترش داده شود؟

# استراتژی‌های جستجو

۳۶

□ استراتژی جستجو. ترتیب **گسترش دادن** گره‌ها.

□ ارزیابی یک استراتژی.

- **کامل بودن**: آیا الگوریتم در صورت وجود راه‌حل، یافتن راه‌حل را تضمین می‌کند؟
- **بهینه بودن**: آیا الگوریتم همواره کم‌هزینه‌ترین راه‌حل (راه‌حل بهینه) را پیدا می‌کند؟
- **پیچیدگی زمانی**: یافتن راه‌حل به چه میزان زمان نیاز دارد؟
- **پیچیدگی حافظه**: برای انجام جستجو به چه میزان حافظه نیاز است؟

□ پارامترهای مهم.

- **b**: حداکثر تعداد فرزندان یک گره در درخت جستجو
- **d**: عمق کم‌عمق‌ترین گره هدف در درخت جستجو
- **m**: حداکثر طول یک مسیر در فضای حالت (ارتفاع درخت جستجو)