

人工知能入門

-探索による人工知能-

Lecture 7

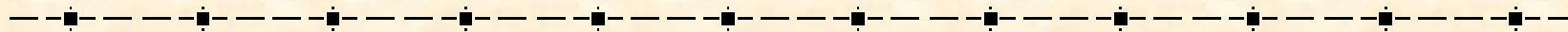
均一コスト探索と深さ優先探索

<http://www2.teu.ac.jp/gamelab/LECTURES/ArtificialIntelligence/index.html>

第1回目ミニテスト

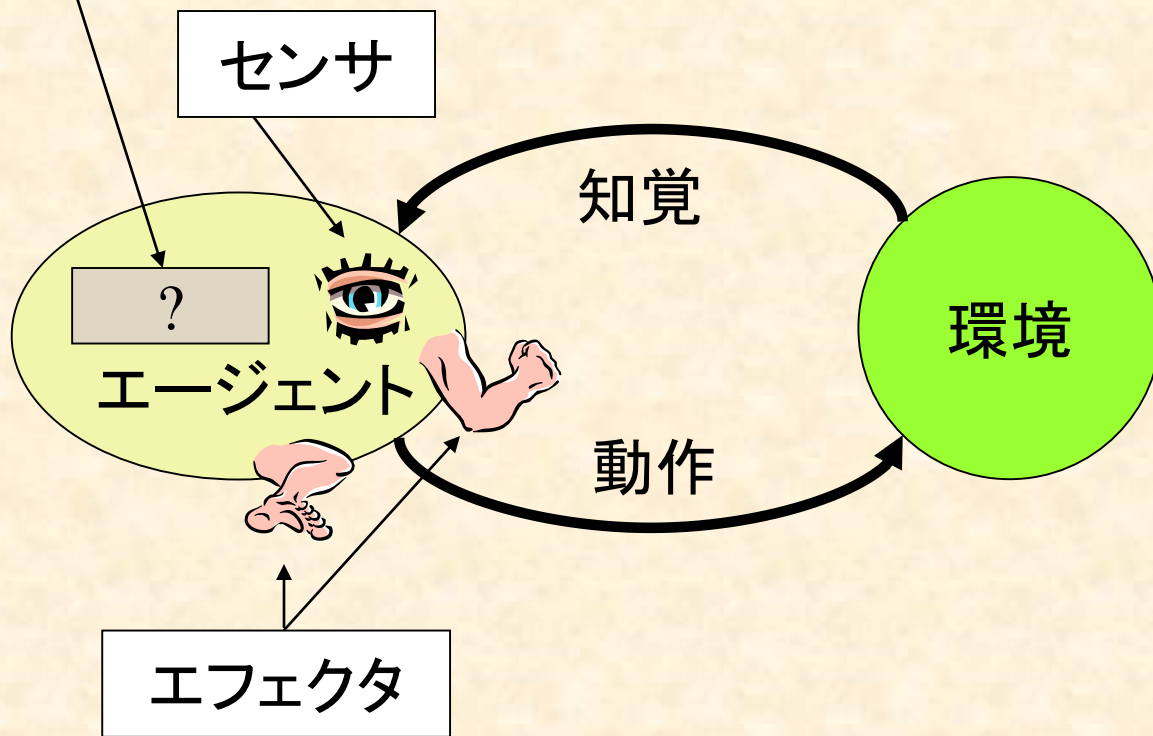
持ち込み可

ミニテストの解説



合理的エージェント

この設計方法は？



一般的探索アルゴリズム

function General-Search(*problem*, *Queuing-Fn*)

returns a solution, or failure

nodes ← Make-Queue(Make-Node(Initial-State[*problem*]))

loop do

if Empty?(*nodes*) **then return** failure

node ← Remove-Front(*nodes*)

if Goal-Test[*problem*] applied to State(*node*) succeeds

then return *node*

nodes ← Queuing-Fn(*nodes*, Expand(*node*, Operators[*problem*]))

end

探索戦略

探索戦略の種類

✦ 情報のない探索 (盲目探索)

- ◆ 現在の状態からのゴールに至るステップの数や経路コストに関する情報を持たない
- ◆ ゴール状態と非ゴール状態との区別することだけ

✦ 情報のある探索 (ヒューリスティック探索)

- ◆ ゴールに近そうな状態を先に展開する等の情報を利用する探索戦略

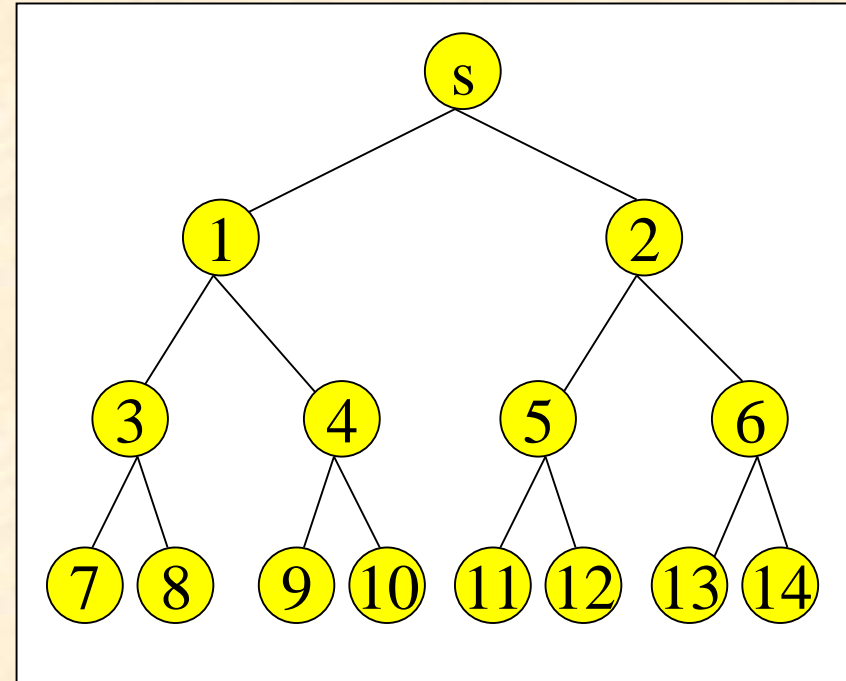
盲目探索

- ✦ 幅優先探索
- ✦ 均一コスト探索
- ✦ 深さ優先探索
- ✦ 深さ制限探索
- ✦ 反復深化探索

盲目探索

幅優先探索

- ✦ まずルートノードを展開
- ✦ 探索木における深さ d のすべてのノードが、深さ $d+1$ のノードの前に展開される
- ✦ 新たに生成された状態を待ち行列の最後に置くことによって実現する



function Breadth-First-Search(*problem*) **returns** a solution or failure
return General-Search(*problem*, Enqueue-At-End)

幅優先探索の評価

★ 幅優先探索の評価

- ◆ 完全性？
- ◆ 時間計算量？
- ◆ 空間計算量？
- ◆ 最適性？

幅優先探索には最適性がない...
最適性を保証する探索戦略は？

盲目探索

- ✦ 幅優先探索
- ✦ 均一コスト探索
- ✦ 深さ優先探索
- ✦ 深さ制限探索
- ✦ 反復深化探索

盲目探索

均一コスト探索

✦ 幅優先探索

- ◆ 最も浅いゴール状態を見つける
- ◆ 最小コストの解と限らない

✦ 均一コスト探索

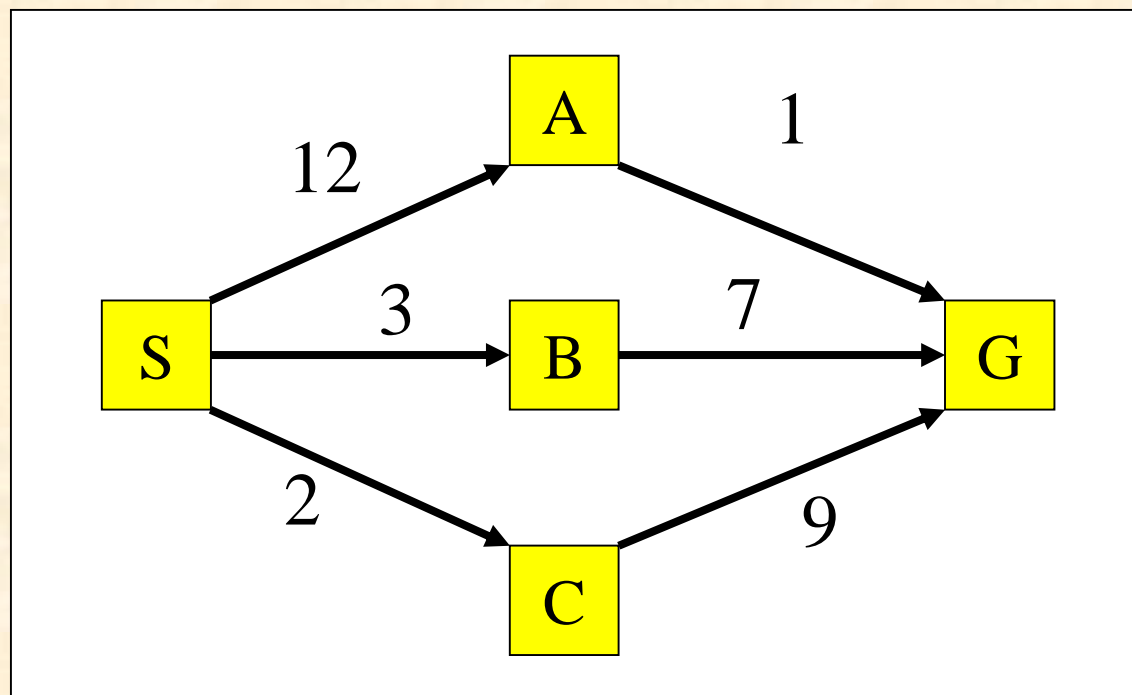
- ◆ 縁におけるコストが最も少ないノードを展開する
- ◆ コストが最も少ないノード: 経路コスト $g(n)$ が一番小さいノード
- ◆ 幅優先探索が $g(n) = \text{Depth}(n)$ の均一コスト探索である
- ◆ **注意**: ゴール検索のタイミングが納得できる!

盲目探索

均一コスト探索

✦ 演習問題7-1

- ◆ 初期状態はS、ゴール状態はGである
- ◆ 数字は各オペレータのコスト
- ◆ 戻るオペレーターはないこと(矢印方向だけ)



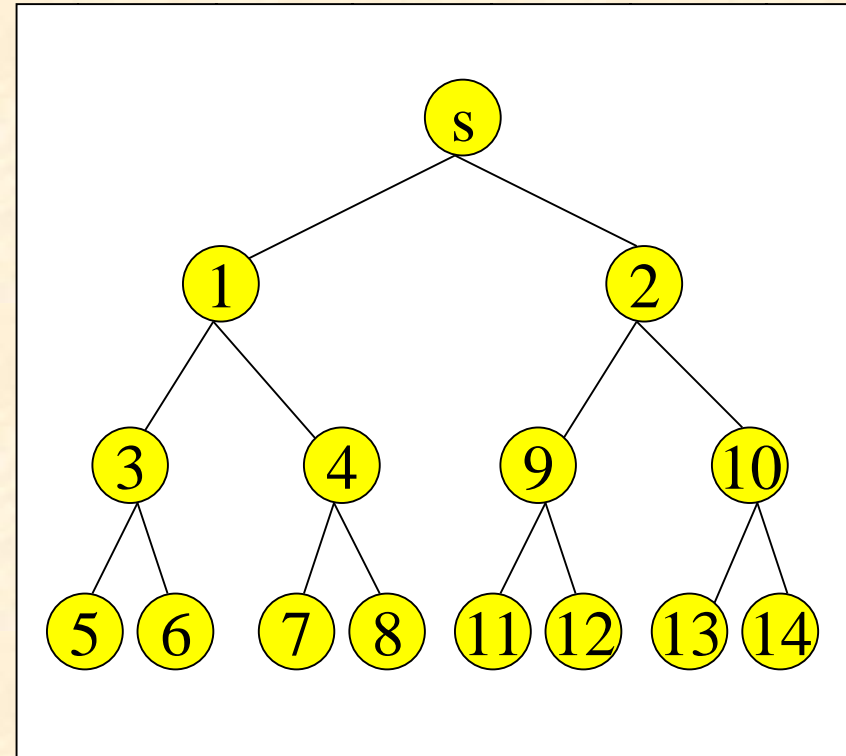
盲目探索

- ✦ 幅優先探索
- ✦ 均一コスト探索
- ✦ 深さ優先探索
- ✦ 深さ制限探索
- ✦ 反復深化探索

盲目探索

深さ優先探索

- ✦ 木の最も深いレベルのノードの一つを展開する
- ✦ 探索が行き止りになるときに限り、探索は後戻りし、より浅いレベルのノードを展開する
- ✦ 最も最後に生成された状態を待ち行列の先頭に置くことによって実現する



function Depth-First-Search(*problem*) **returns** a solution or failure

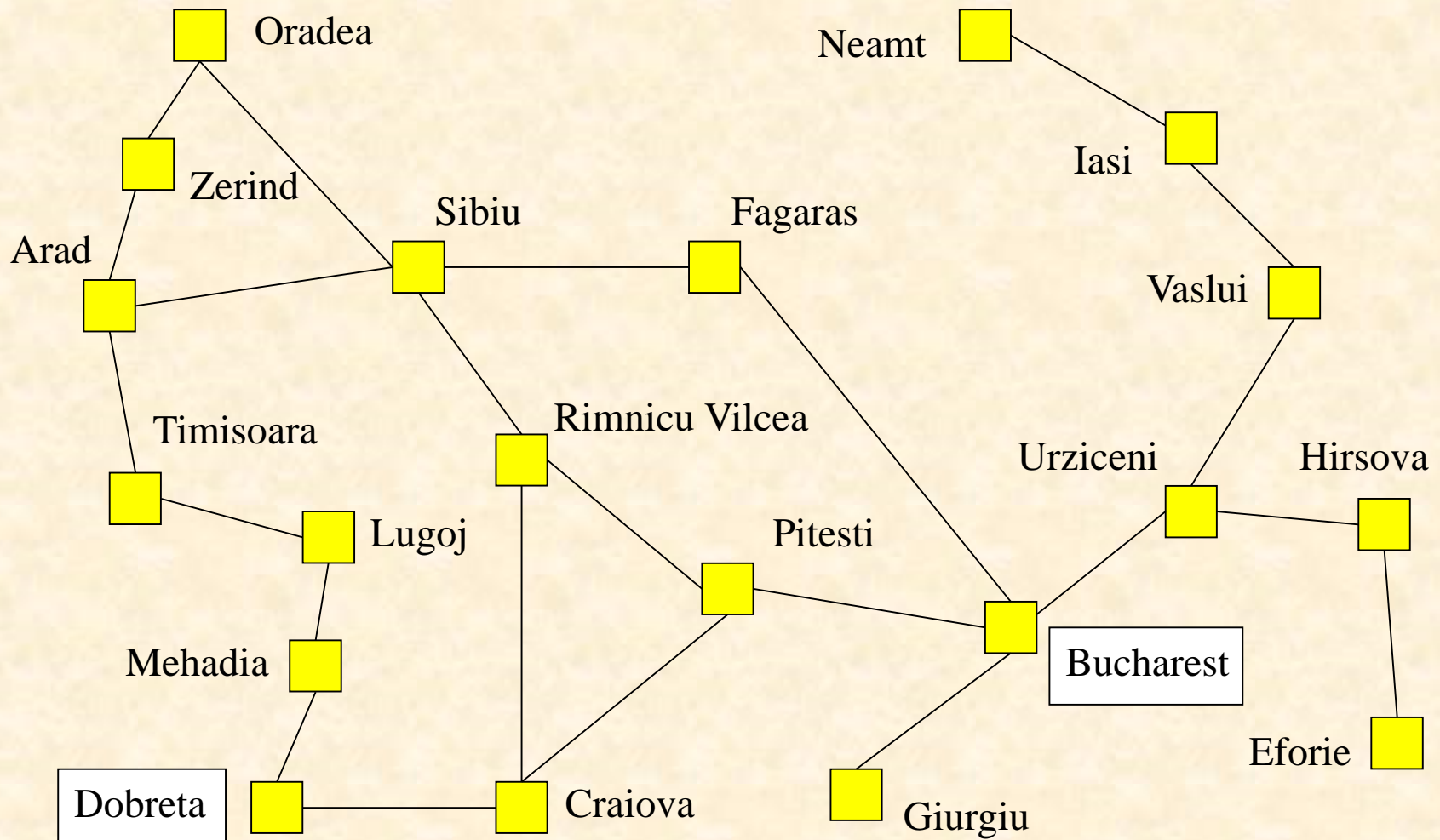
General-Search(*problem*, Enqueue-At-Front)

深さ優先探索の展開

| 展開回数 | 待ち行列 | 展開されたノード |
|------|-----------|---------------|
| 0 | [s] | ∅ |
| 1 | [1 2] | [s] |
| 2 | [3 4 2] | [1 s] |
| 3 | [5 6 4 2] | [3 1 s] |
| 4 | [6 4 2] | [5 3 1 s] |
| 5 | [4 2] | [6 5 3 1 s] |
| 6 | [7 8 2] | [4 6 5 3 1 s] |
| : | : | : |

深さ優先探索

演習問題7-2: DobretaからBucharestへ



盲目探索

深さ優先探索

★ 深さ優先探索のメリット

- ◆ **メモリ必要量は適度**: ルートから葉ノードへの単一の経路と、その経路の各ノードに対するまだ展開されていない兄弟ノードだけを格納する必要がある
 - 分枝度 b で最大の深さが m の状態空間に対して、深さ優先探索は bm 個のノードだけを格納すればよい(幅優先探索: b^d)
 - 深さ $d = 12$ において111テラバイトの代わりに12キロバイトしか必要とせず、100億分の1の空間ですむ
- ◆ **時間計算量**
 - 非常に多くの解がある問題に対しては、深さ優先は幅優先よりも実際には早いかもしれないが、最悪の場合では $O(b^m)$

★ 深さ優先探索の欠点

- ◆ 誤った経路をたどって、行き詰まる可能性はある
- ◆ 無限に深い探索木における不運な選択から回復できない

まとめ

★ 幅優先探索

- ◆ メリット: 完全性がある
- ◆ 問題: 最適性はない、計算量

★ 均一コスト探索

- ◆ メリット: 完全性と最適性がある
- ◆ 問題: 計算量

★ 深さ優先探索

- ◆ メリット: メモリ必要量は適度、解が多い場合にすぐ終わる
- ◆ 問題: 完全性と最適性がない