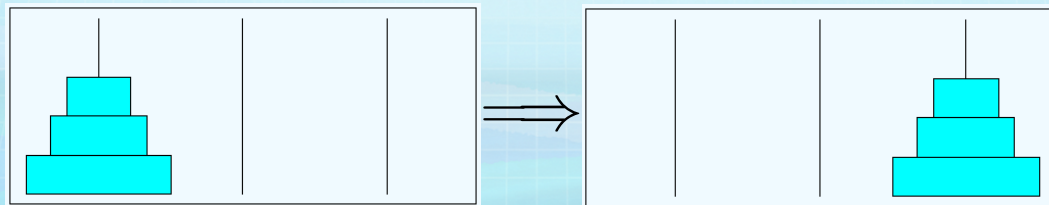


状態空間モデル

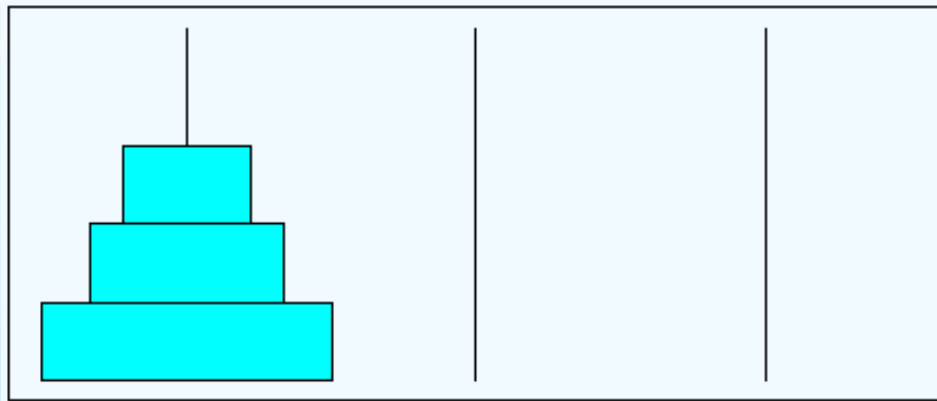
◆ハノイの塔

- 3本の柱と大きさの異なる複数の円盤
- 最初はすべての円盤が左端の柱(大きさの順)
- 円盤を一回に一枚ずつ別の柱に移動させる
- すべての円盤を右端の柱に大きさの順に置く
- 小さな円盤の上に大きな円盤を置くことはできない



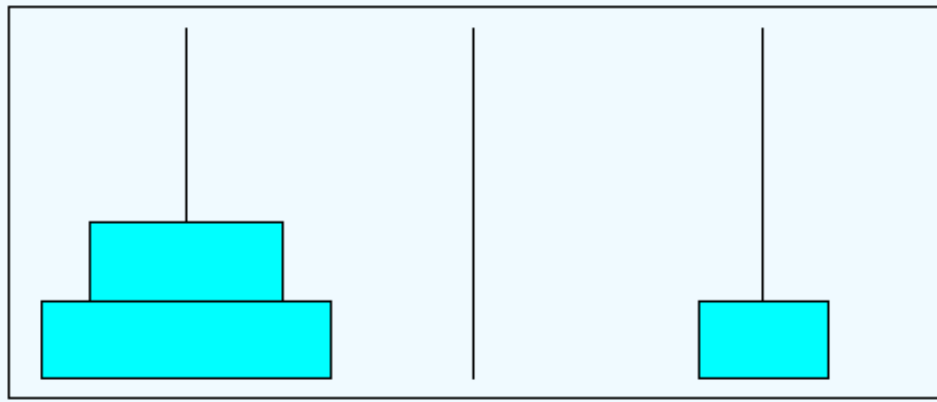
状態空間モデル

◆ 0



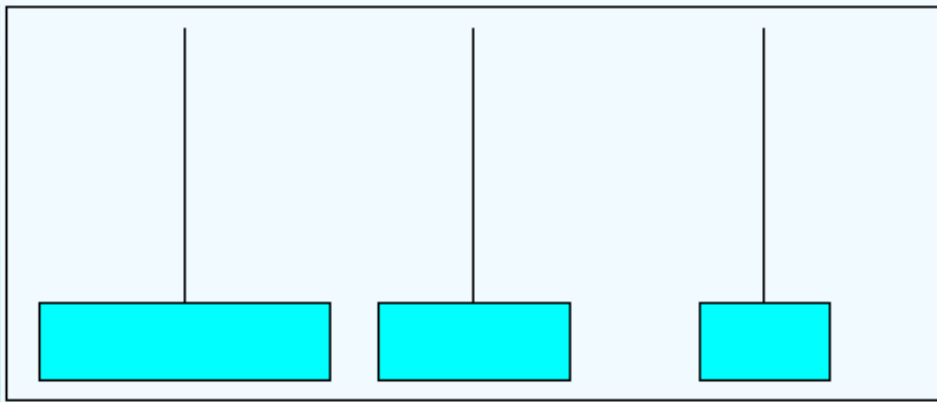
状態空間モデル

◆ 1



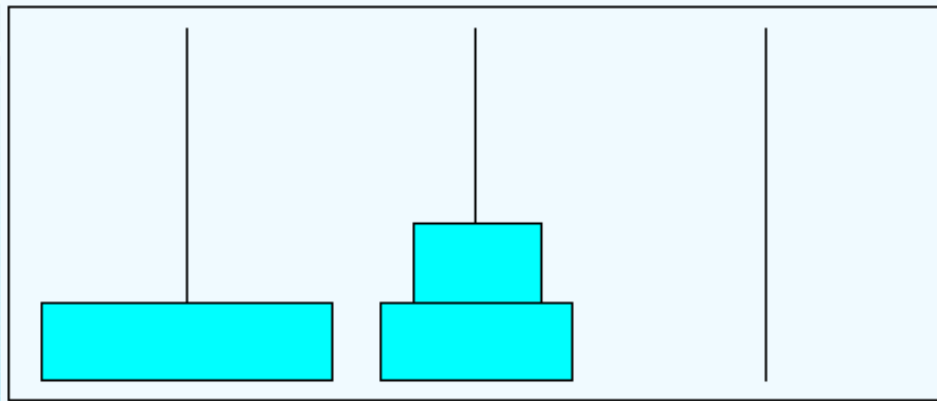
状態空間モデル

◆ 2



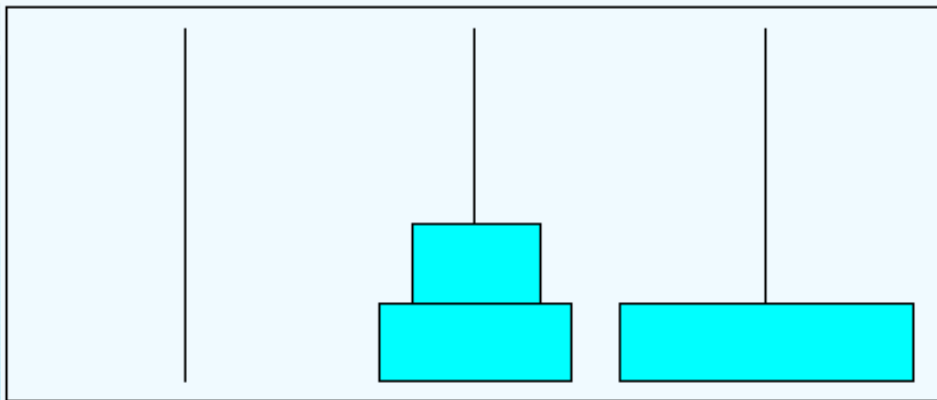
状態空間モデル

◆ 3



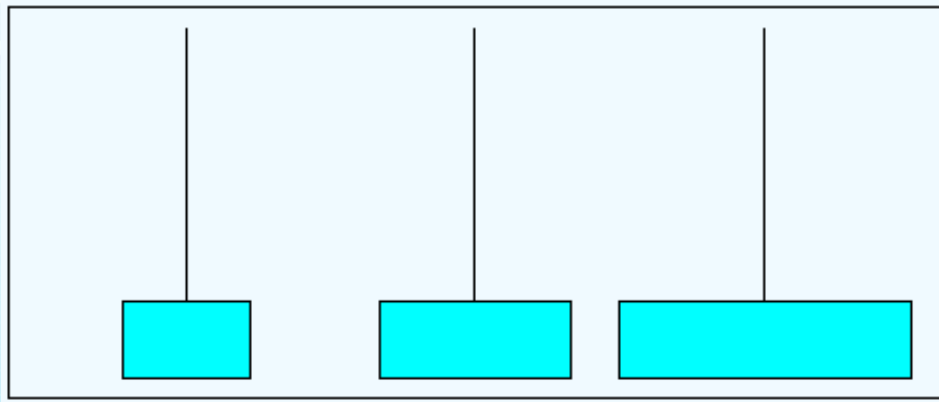
状態空間モデル

◆ 4



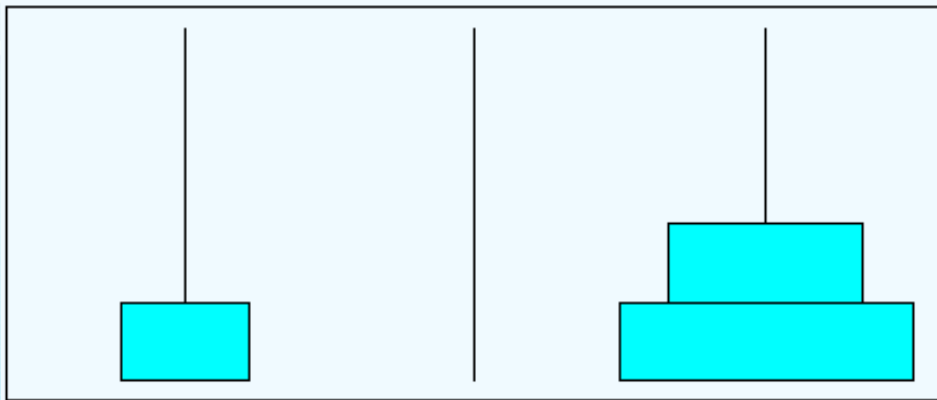
状態空間モデル

◆ 5



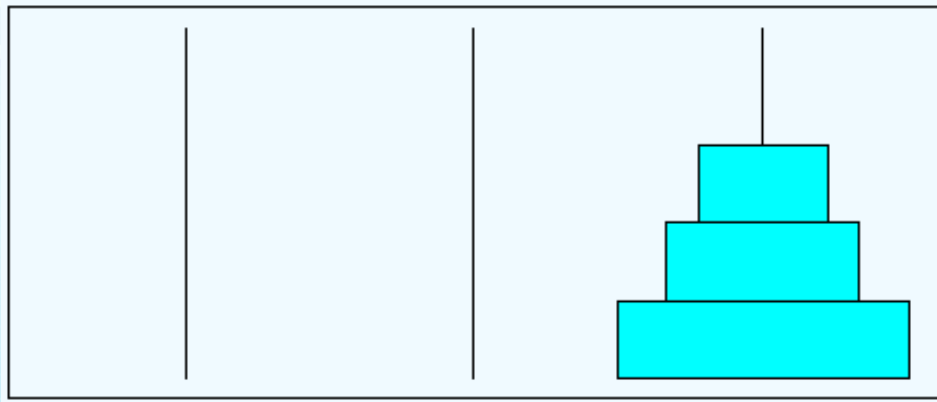
状態空間モデル

◆ 6



状態空間モデル

◆ 7



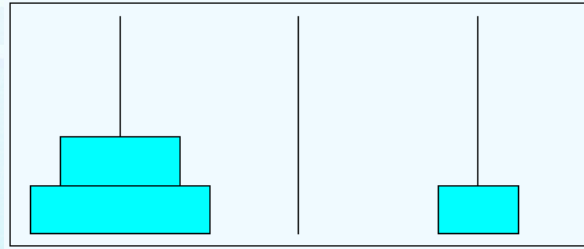
状態空間モデル

◆問題

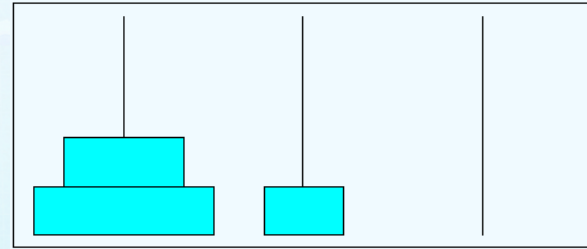
- 状態の集合
- 作用素の集合
- 初期状態
- 目標状態の集合

状態空間モデル

状態 問題が解かれていく各ステップの状況



(L, L, R)



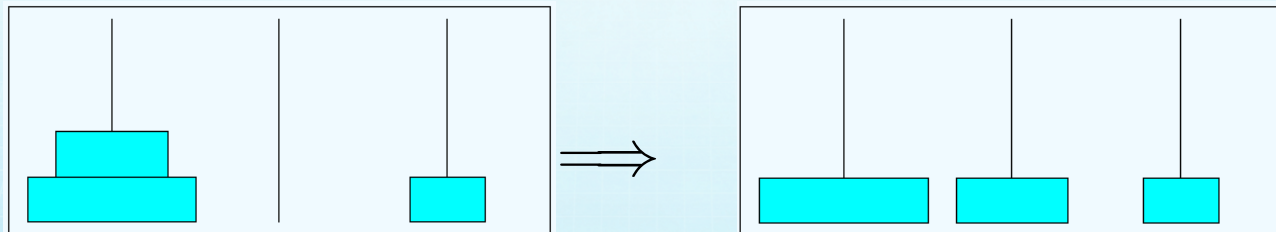
(L, L, C)

状態の集合 とり得るすべての状態

状態空間モデル

作用素（オペレータ） ある状態から別の状態へ遷移するための手段

- 左の柱にある一番上の円盤を中央の柱に移動



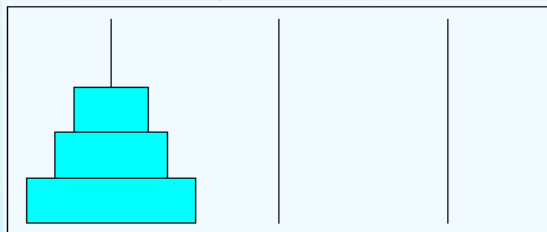
$(L, L, R) \rightarrow (L, C, R)$, $(L, L, R) - LC \rightarrow (L, C, R)$
 $LC(x)$, $LR(x)$, $CL(x)$, $CR(x)$, $RL(x)$, $RC(x)$

作用素の集合 とり得るすべての状態遷移

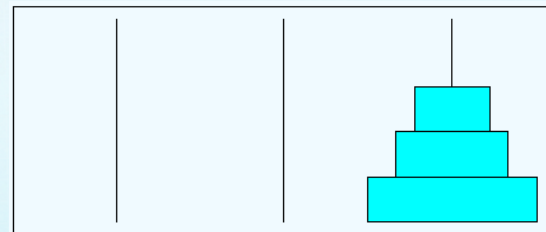
状態空間モデル

初期状態 最初に与えられた状態

目標状態 達成すべき状態 (ゴール)



初期状態



目標状態

初期状態 1個

目標状態 1個ないしは複数

状態空間モデル

問題 (状態の集合, 作用素の集合, 初期状態, 目標状態の集合)

状態 問題が解かれていく各ステップの状況

作用素 (オペレータ) ある状態から別の状態へ遷移するための手段

初期状態 最初に与えられた状態

目標状態 達成すべき状態 (ゴール)

状態空間モデル

◆ (ある種の) 定理証明

初期状態 前提となる数式

目標状態 定理に相当する数式

作用素 式変形

状態 式変形の過程の数式

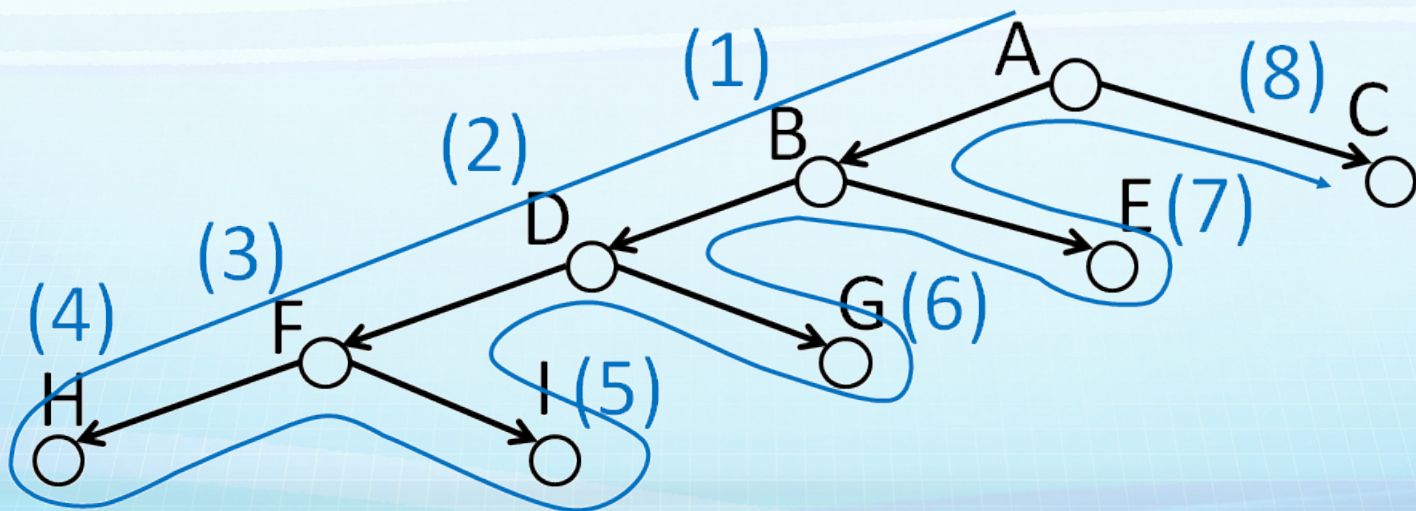
系統的探索

系統的探索

- 問題に関する知識を用いずシラミ潰しに探索
- 汎用的
- 原理的には解（目標）の発見が保証される

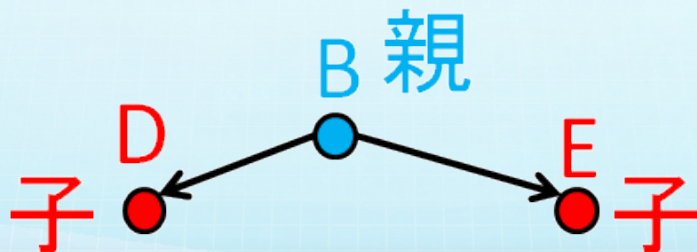
系統的探索

◆ 深さ優先探索 (縦型探索)



系統的探索

- 有向グラフにおける
 - 枝の始点 … 親
 - 枝の終点 … 子



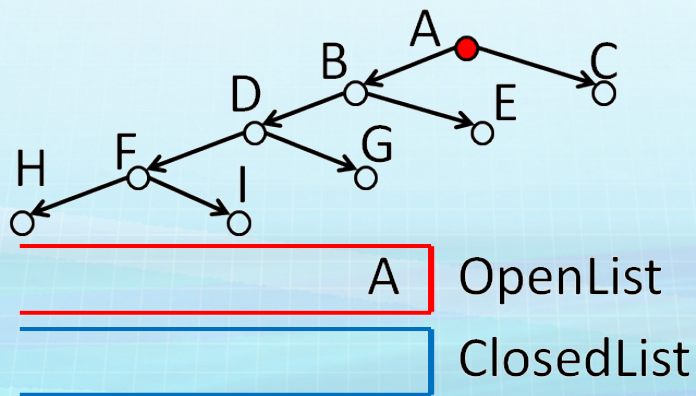
系統的探索

◆ 深さ優先探索

- (1) 探索を開始する点を OpenList に入れる
ClosedList は空にする
- (2) OpenList が空 (解がない) なら探索は失敗して終了
- (3) OpenList の先頭の点 n を取り除き, ClosedList に入れる
- (4) n が目標点であるなら探索は成功して終了
- (5) n の子のうち, OpenList や ClosedList に含まれていない点はすべてを OpenList の先頭に入れて (2) へ
そのような点がなければ, そのまま (2) へ

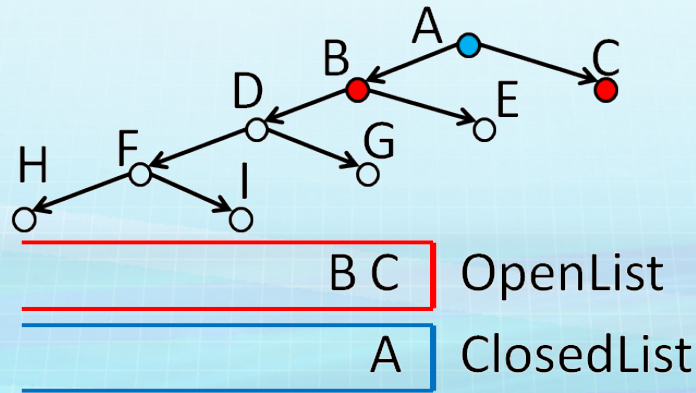
系統的探索

- (1) 探索を開始する点を OpenList に入れる
ClosedList は空にする



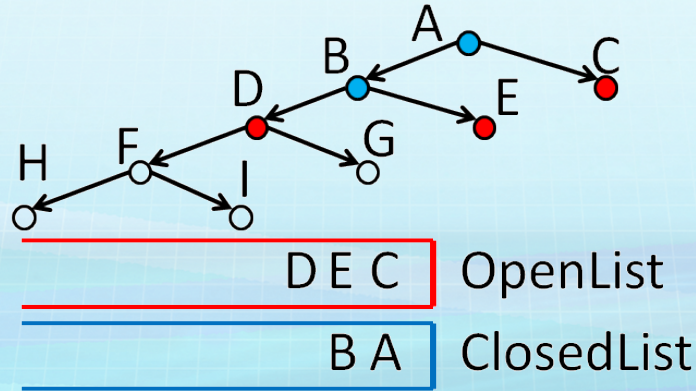
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



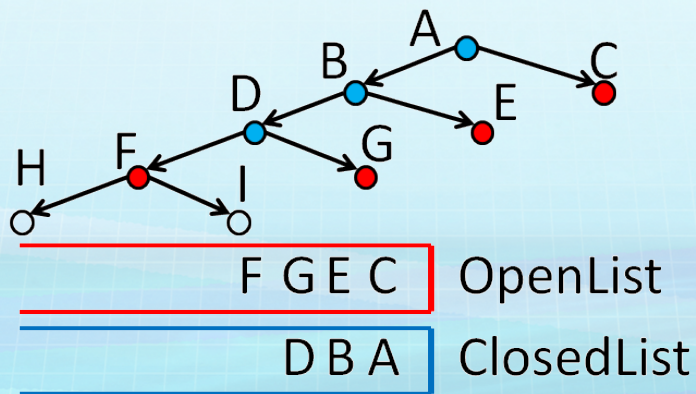
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



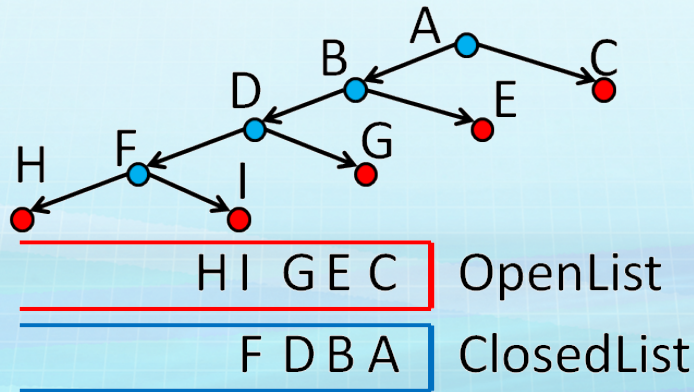
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



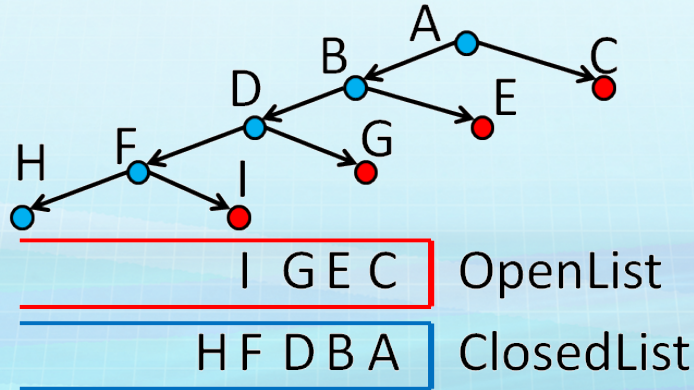
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



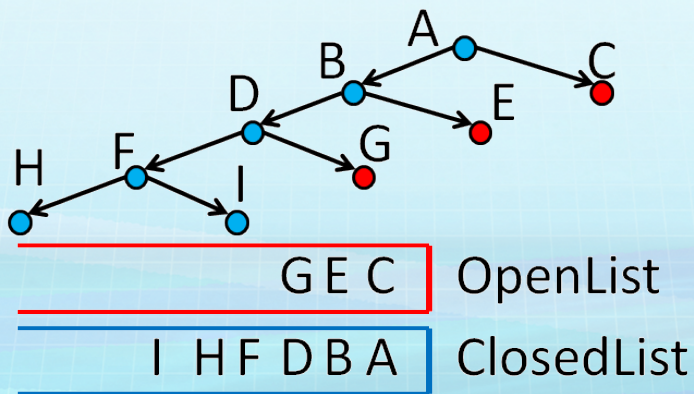
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



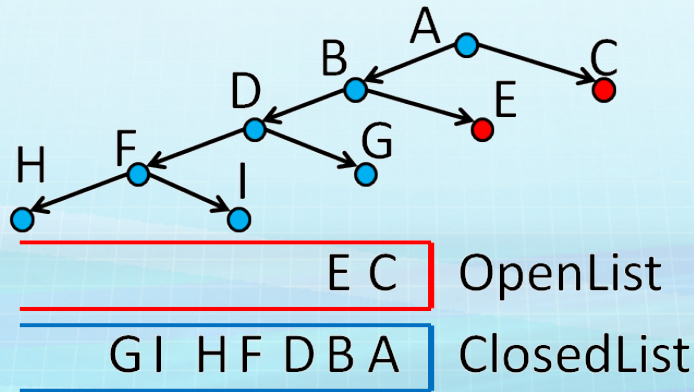
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



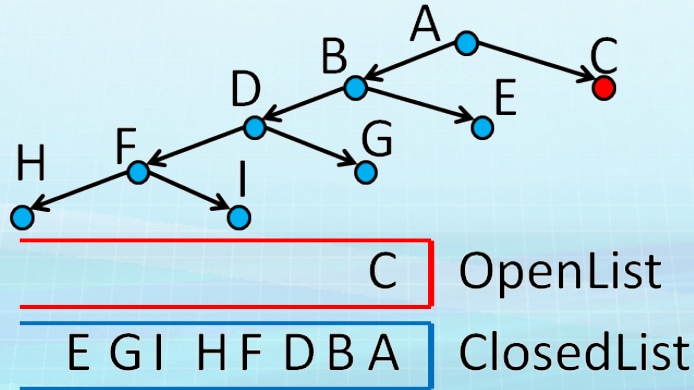
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



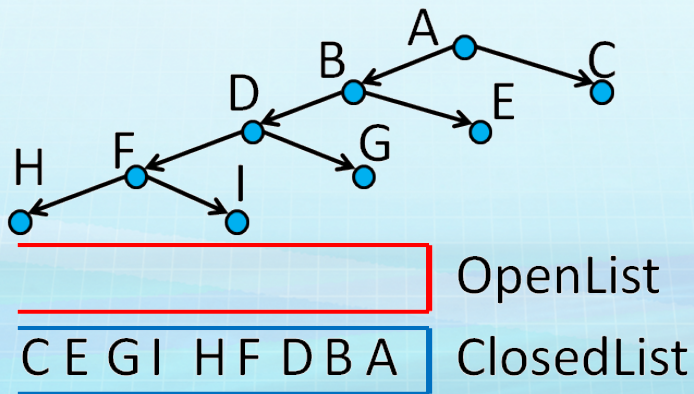
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの先頭に入れて(2)へ
そのような点がなければ, そのまま(2)へ



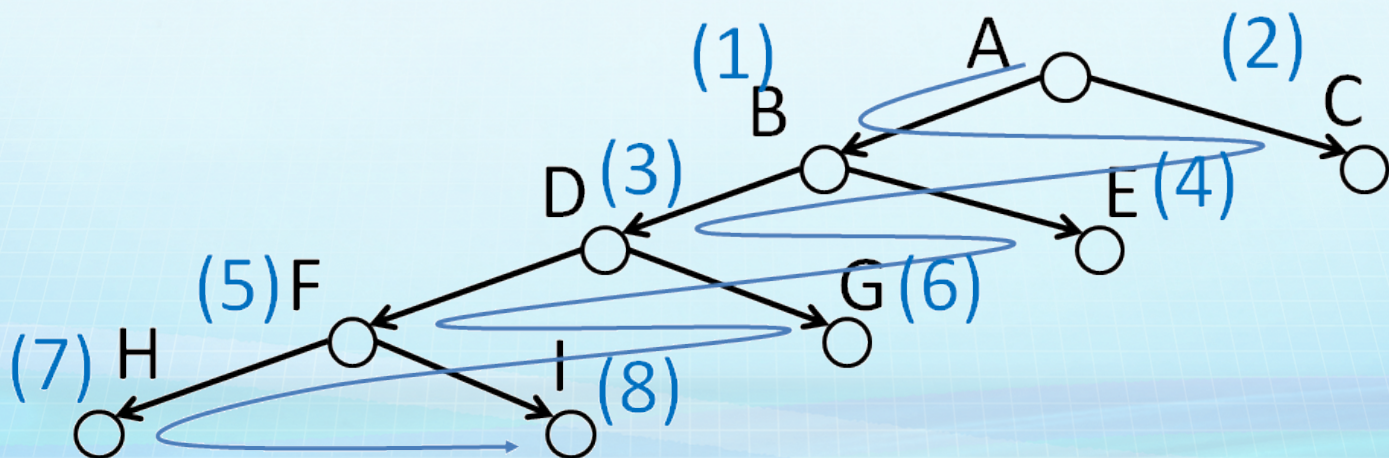
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (4) n が目標点であるなら探索は成功して終了



系統的探索

◆ 幅優先探索 (幅型探索)



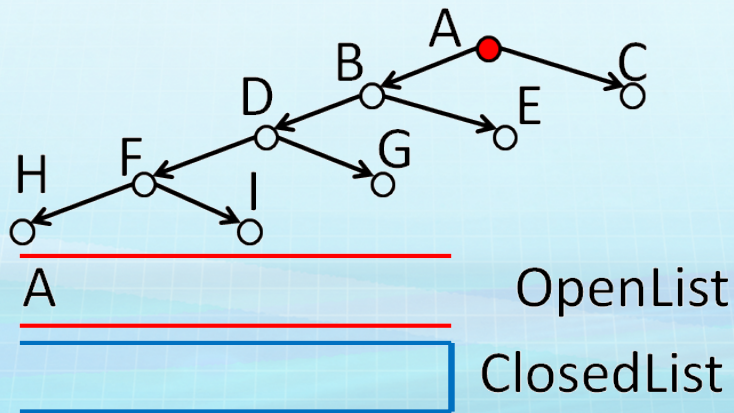
系統的探索

◆幅優先探索

- (1) 探索を開始する点を OpenList に入れる
ClosedList は空にする
- (2) OpenList が空 (解がない) なら探索は失敗して終了
- (3) OpenList の先頭の点 n を取り除き, ClosedList に入れる
- (4) n が目標点であるなら探索は成功して終了
- (5) n の子のうち, OpenList や ClosedList に含まれていない点はすべてを OpenList の末尾に入れて (2) へ
そのような点がなければ, そのまま (2) へ

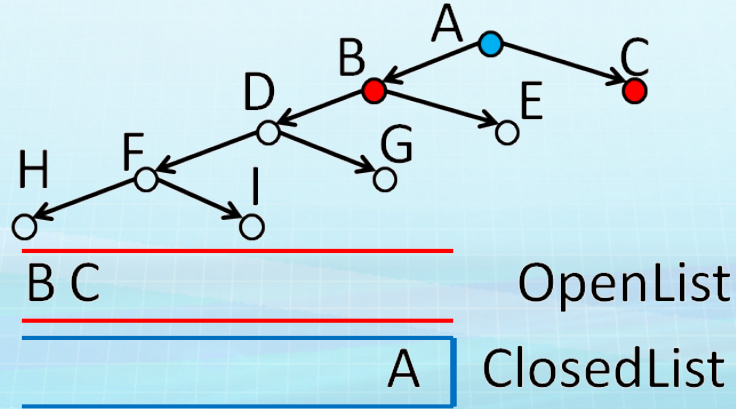
系統的探索

- (1) 探索を開始する点を OpenList に入れる
ClosedList は空にする



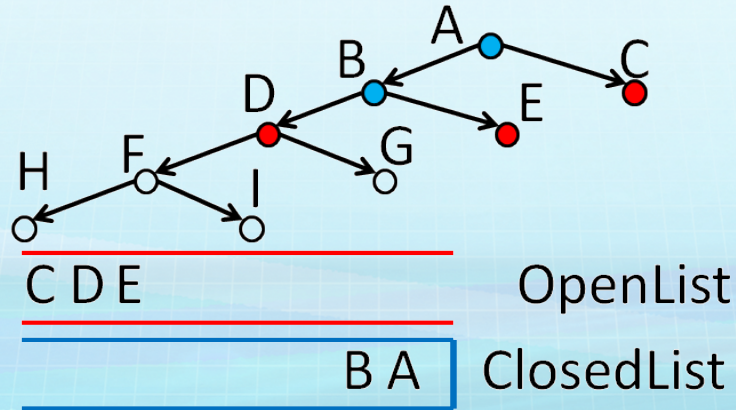
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



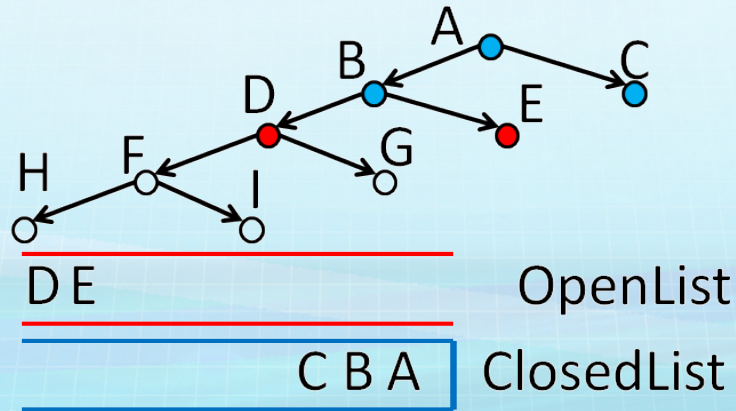
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



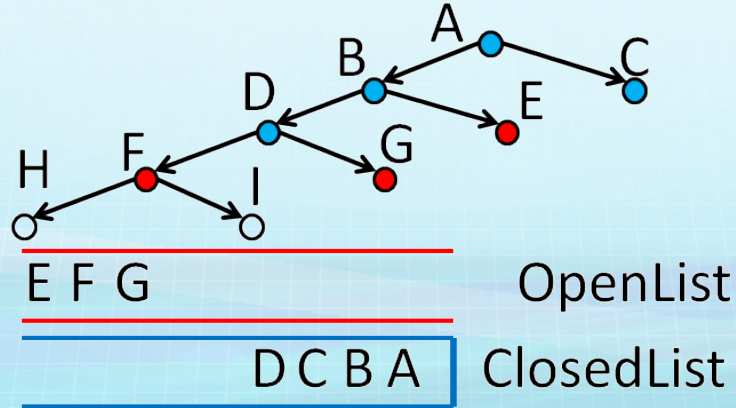
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



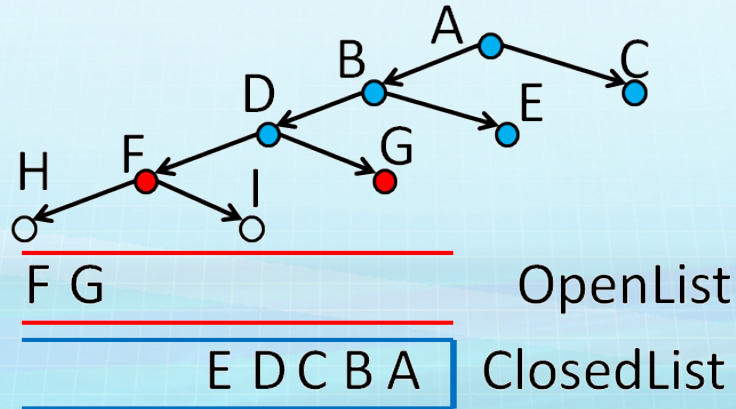
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



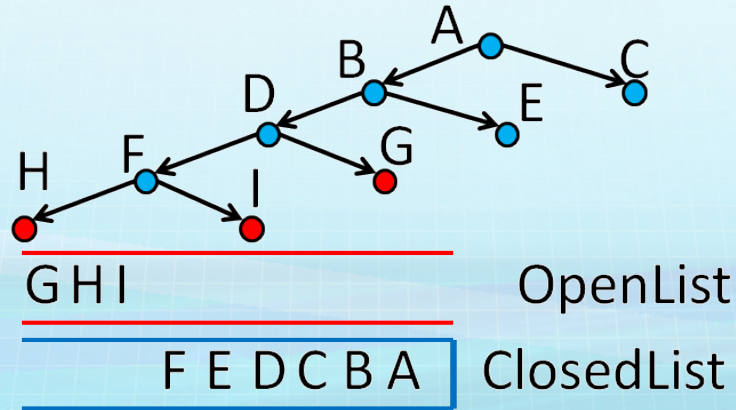
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



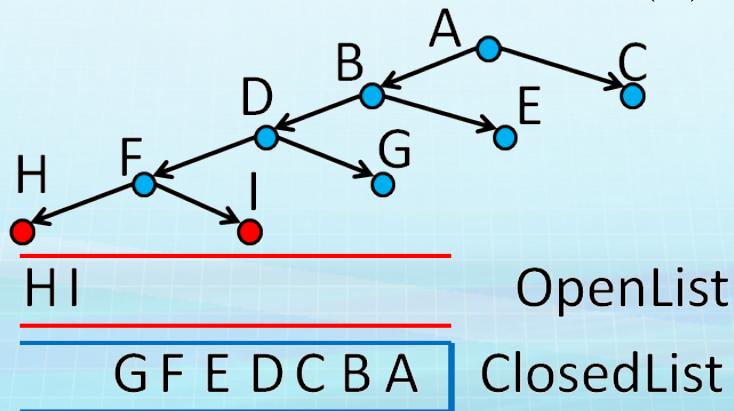
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



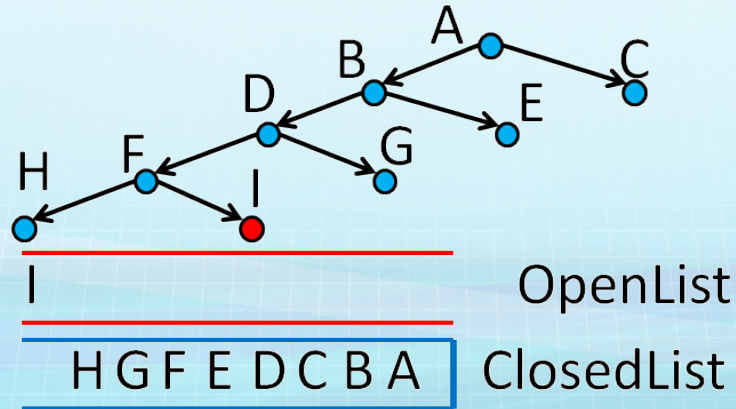
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



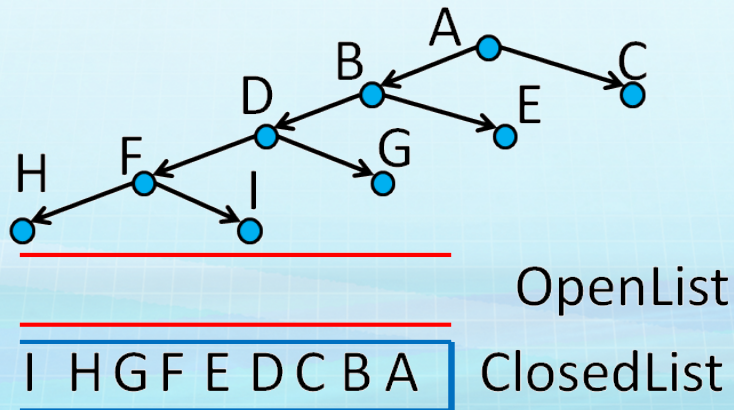
系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (5) n の子のうち, OpenListやClosedListに含まれていない点はすべてをOpenListの末尾に入れて(2)へ
そのような点がなければ, そのまま(2)へ



系統的探索

- (3) OpenListの先頭の点 n を取り除き, ClosedListに入れる
- (4) n が目標点であるなら探索は成功して終了



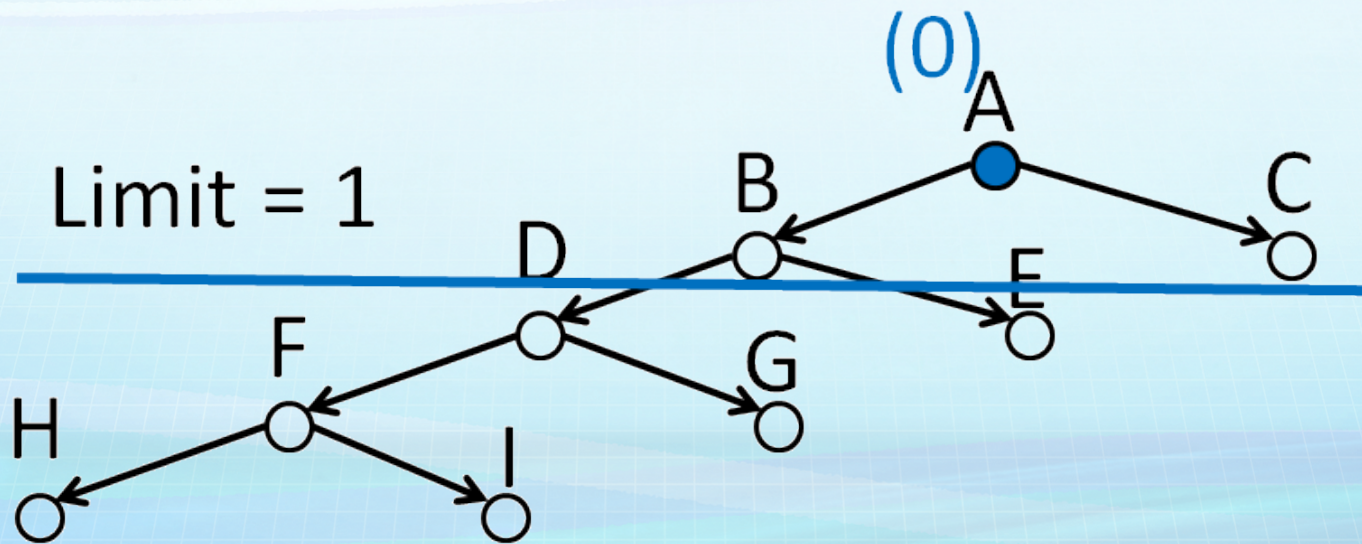
系統的探索

◆反復深化深さ優先探索

- (0) 探索する深さの制限 $Limit$ を 1 とする
- (1) 探索を開始する点を $OpenList$ に入れ, $ClosedList$ は空にする
- (2) $OpenList$ が空なら $Limit$ を 1 大きくし, (1) へ
- (3) $OpenList$ の先頭の点 n を取り除き, $ClosedList$ に入れる
- (4) n が目標点であるなら探索は成功して終了
- (5) n の深さが $Limit$ より小さい場合, n の子のうち, $OpenList$ や $ClosedList$ に含まれていない点はすべてを $OpenList$ の先頭に入れて (2) へ. それ以外の場合は, そのまま (2) へ

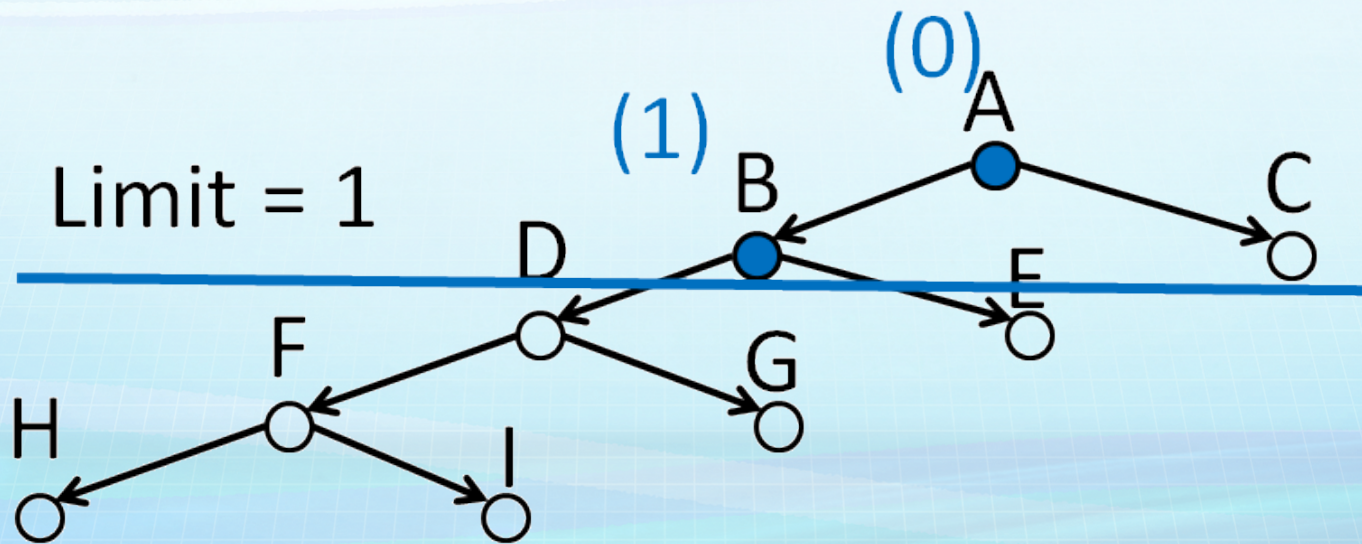
系統的探索

◆ 反復深化深さ優先探索



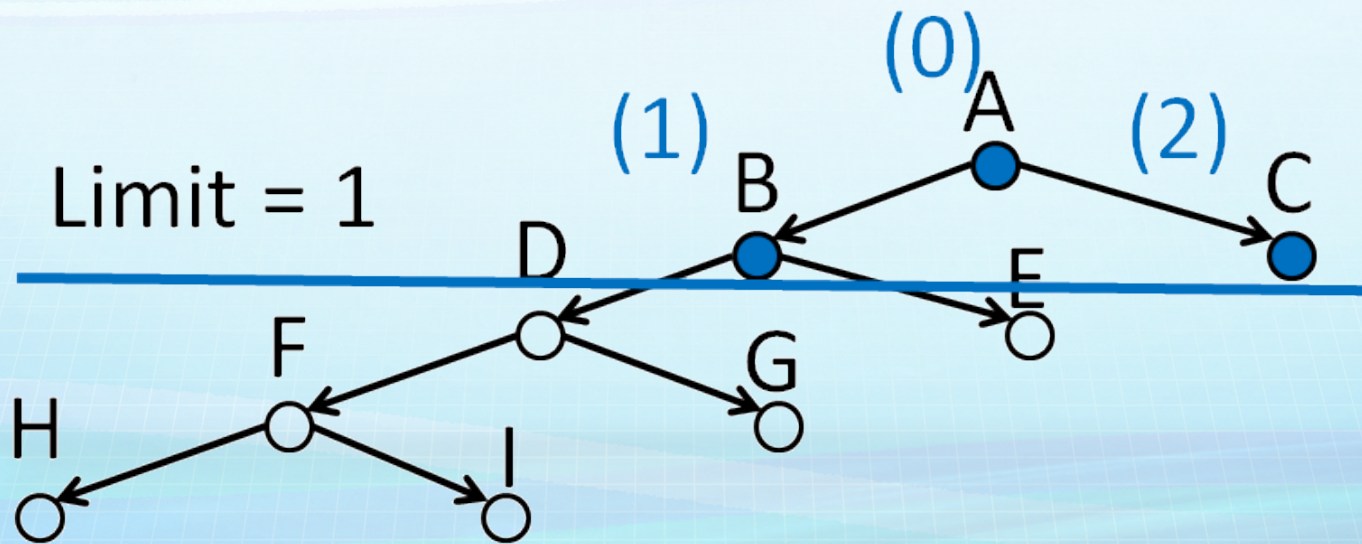
系統的探索

◆ 反復深化深さ優先探索



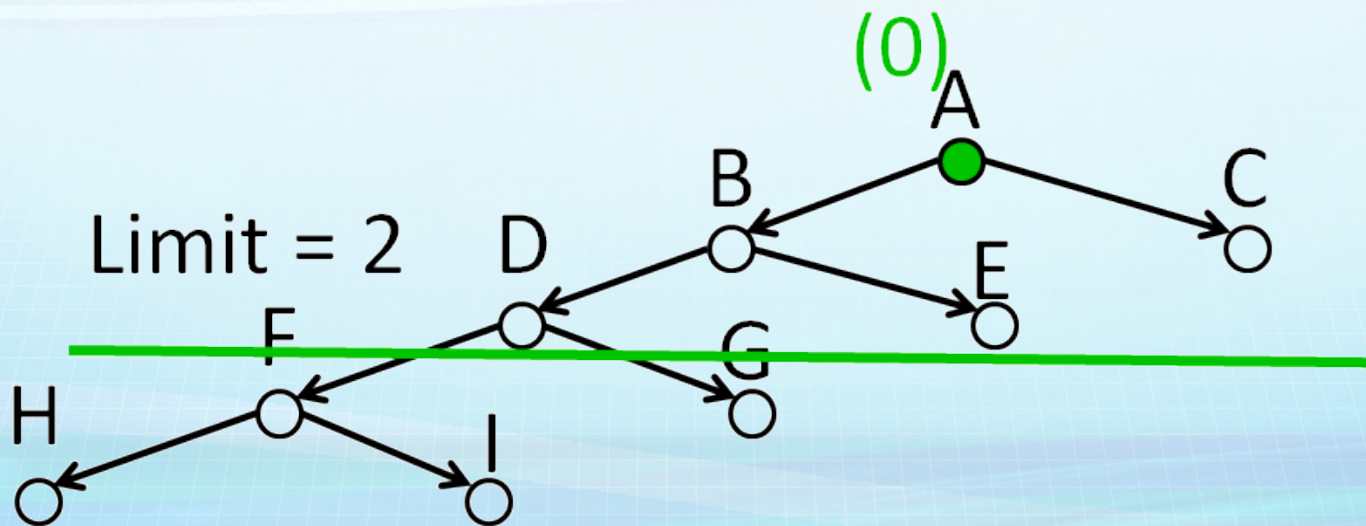
系統的探索

◆反復深化深さ優先探索



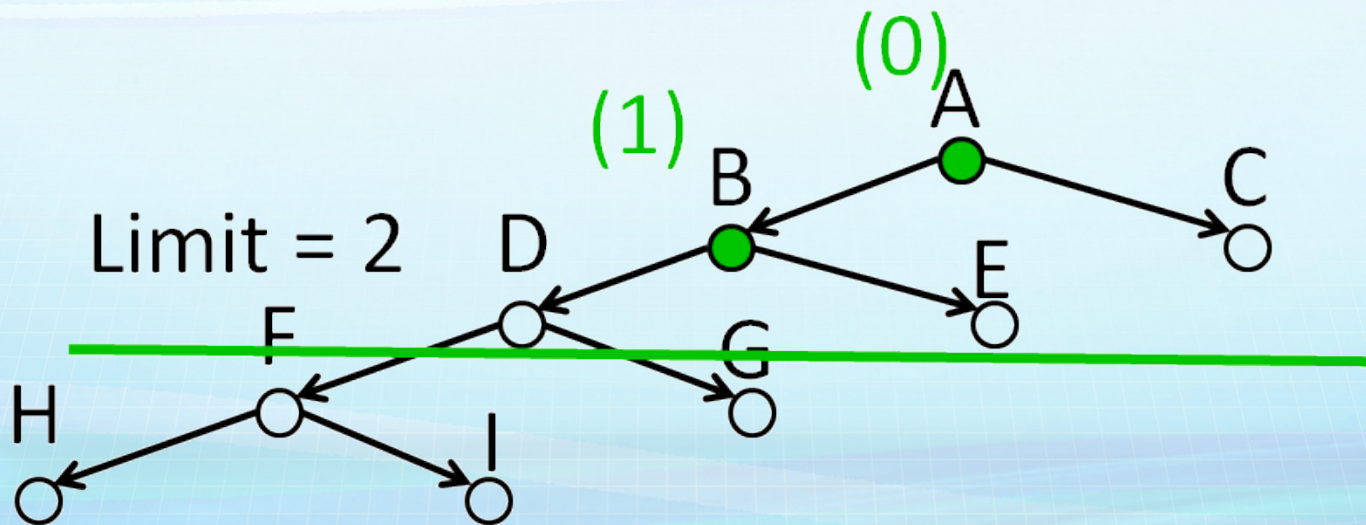
系統的探索

◆ 反復深化深さ優先探索



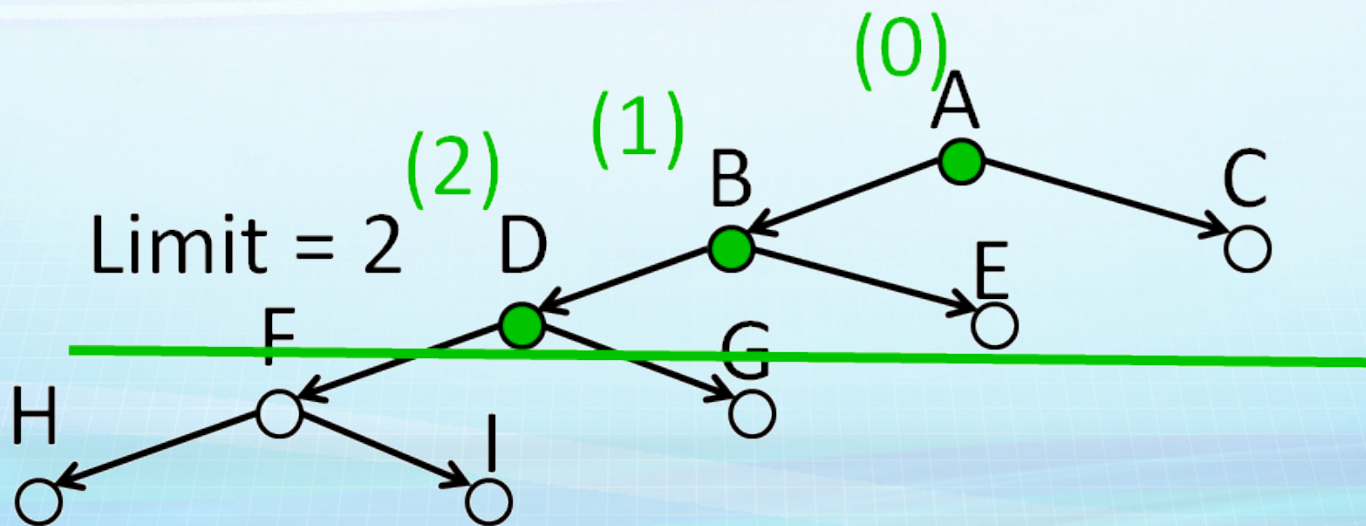
系統的探索

◆反復深化深さ優先探索



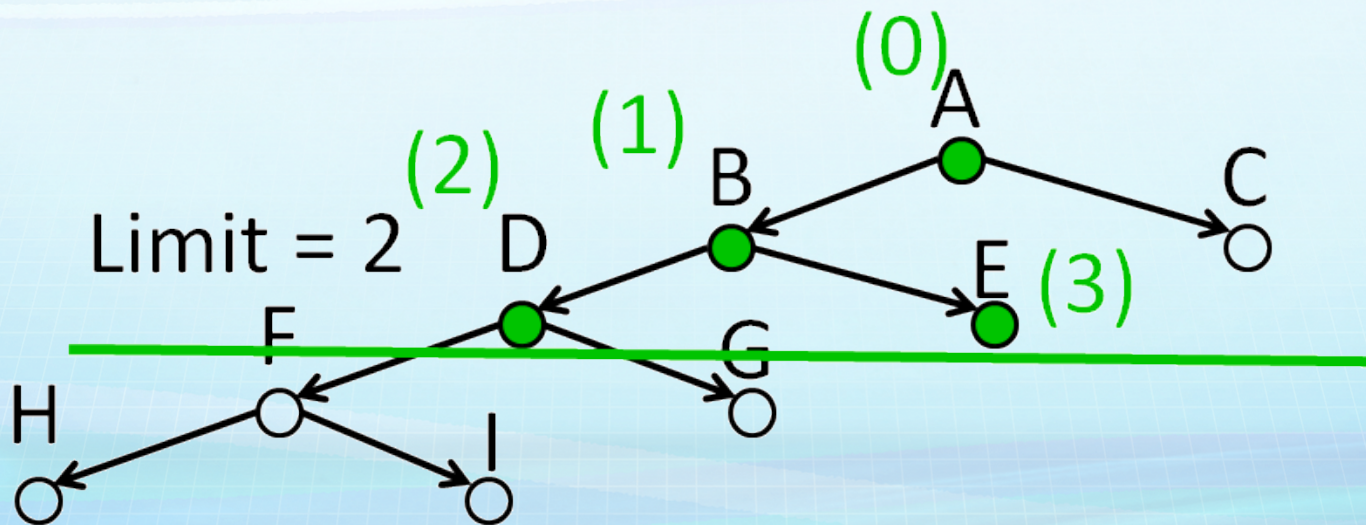
系統的探索

◆ 反復深化深さ優先探索



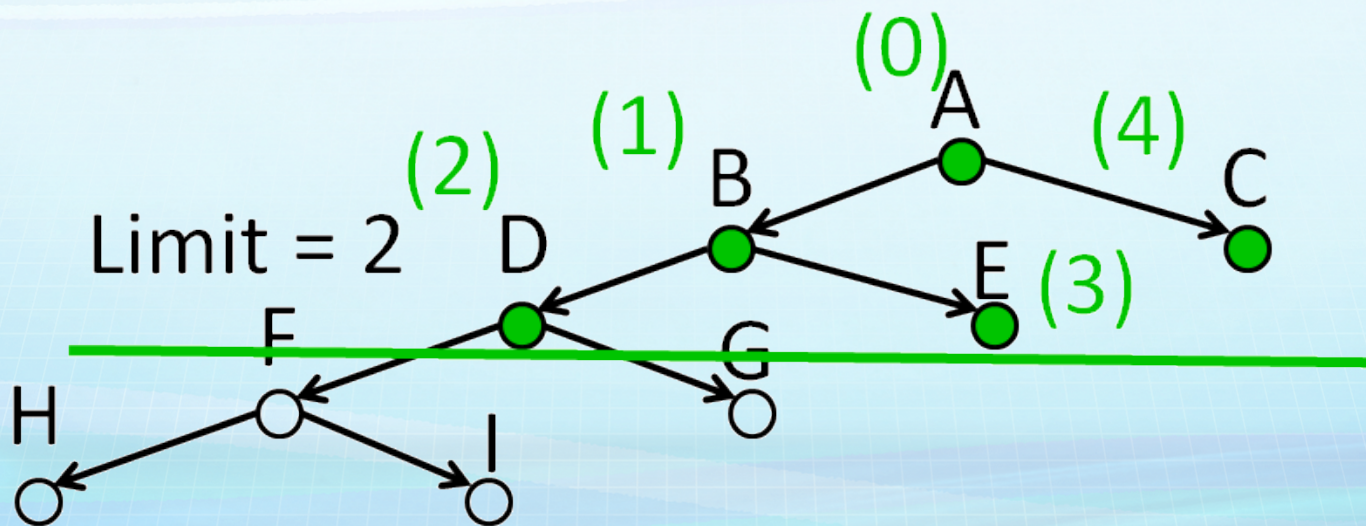
系統的探索

◆ 反復深化深さ優先探索



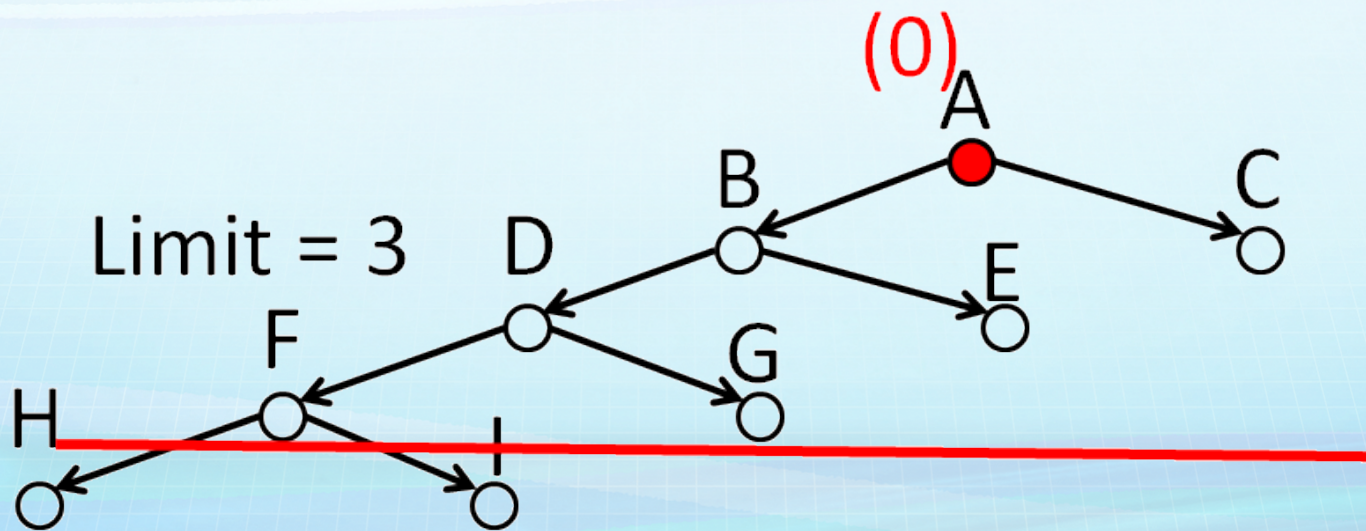
系統的探索

◆ 反復深化深さ優先探索



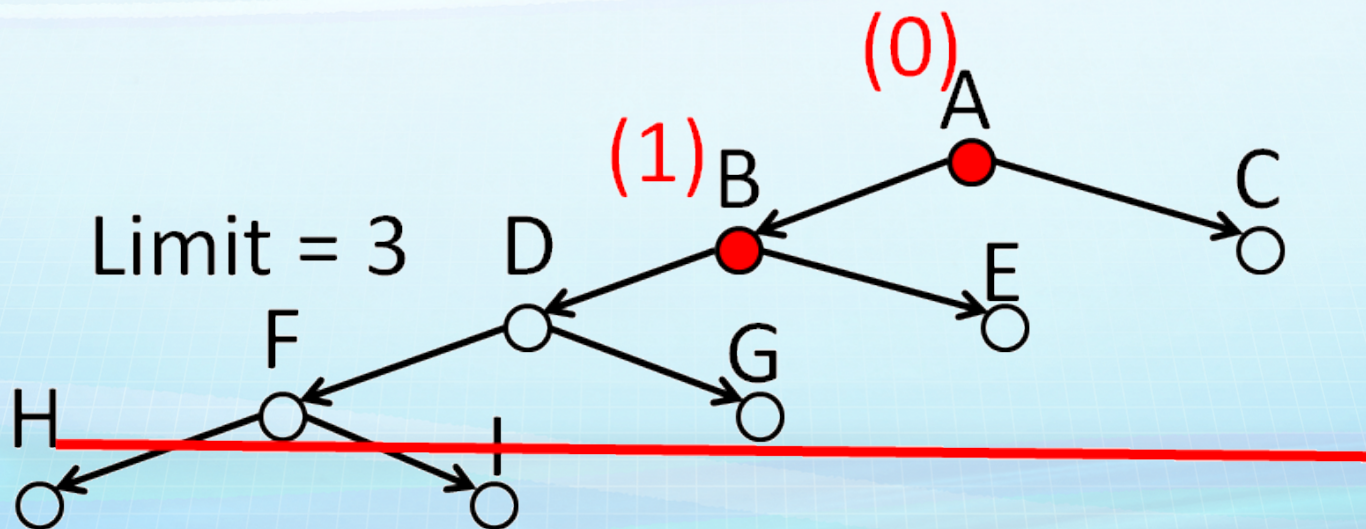
系統的探索

◆反復深化深さ優先探索



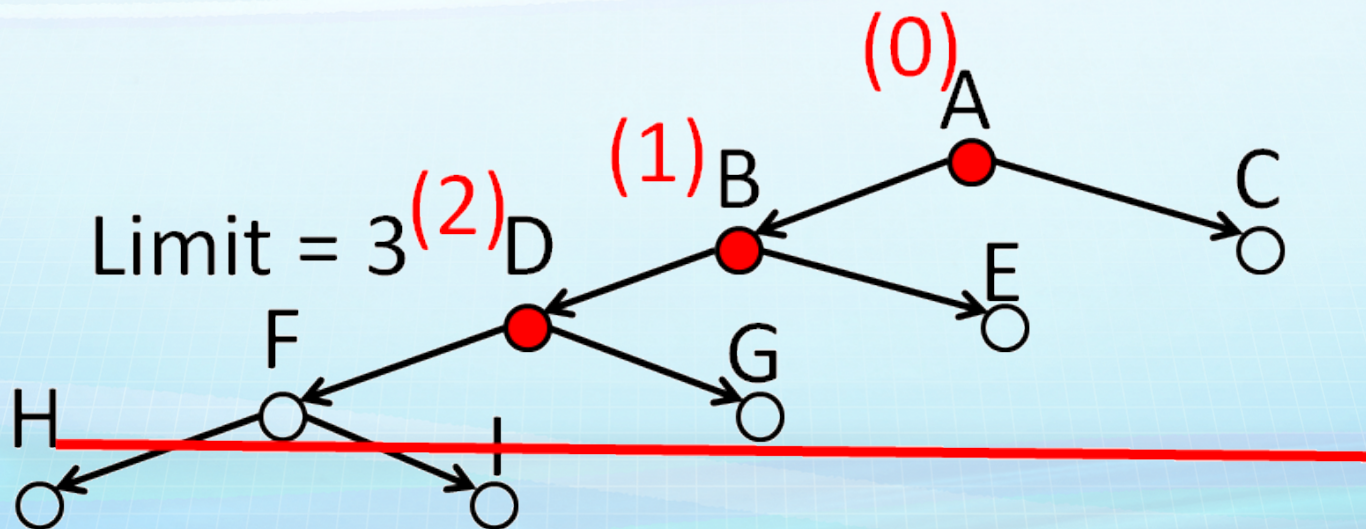
系統的探索

◆ 反復深化深さ優先探索



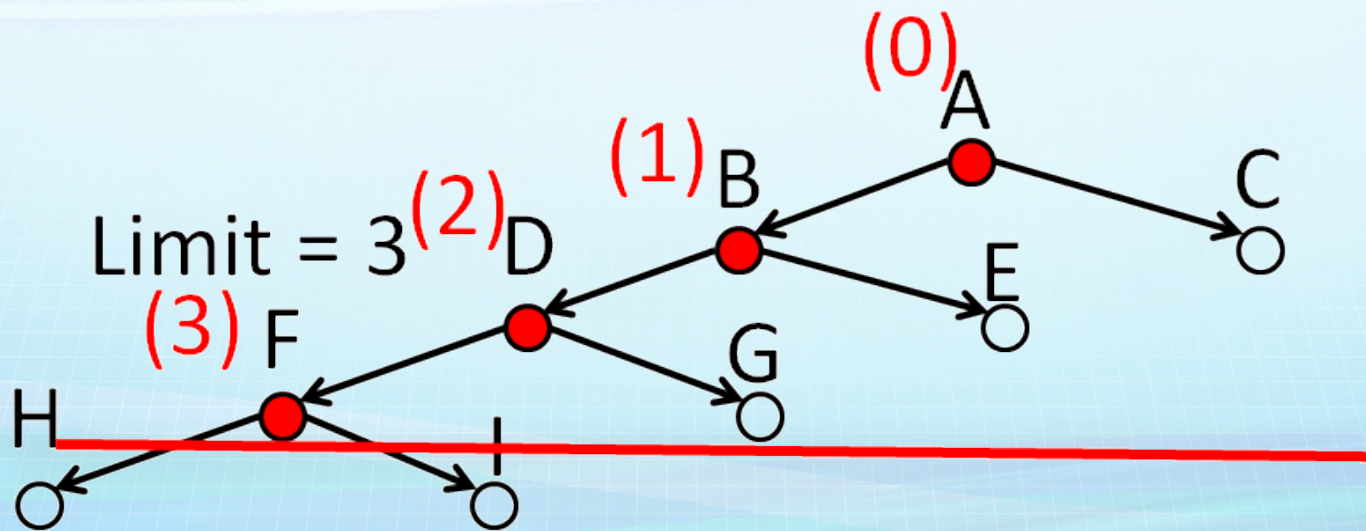
系統的探索

◆ 反復深化深さ優先探索



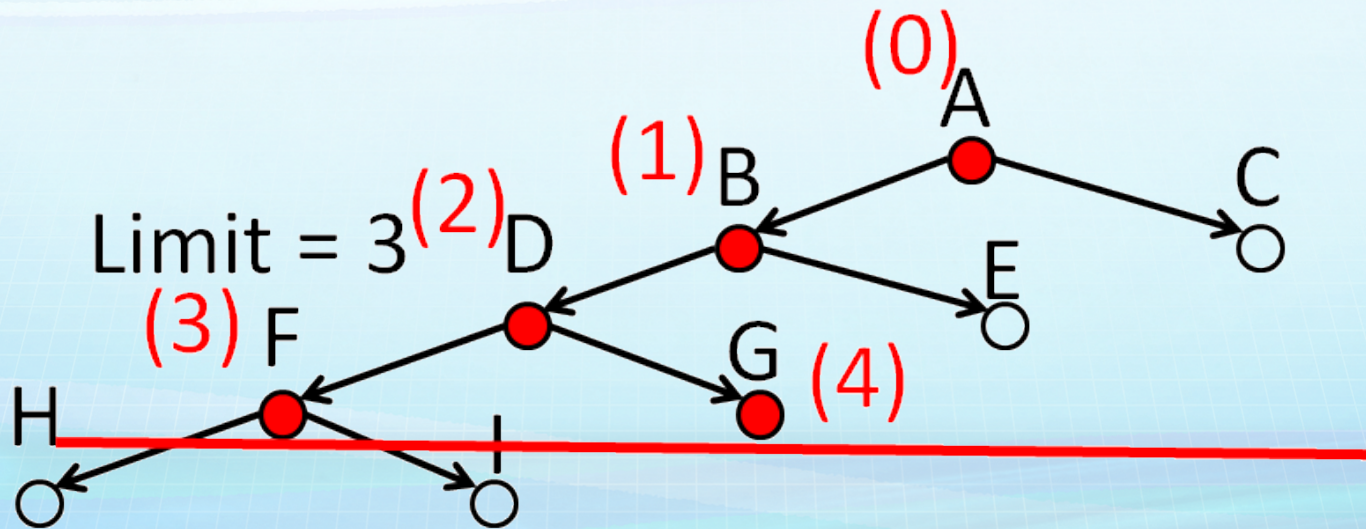
系統的探索

◆ 反復深化深さ優先探索



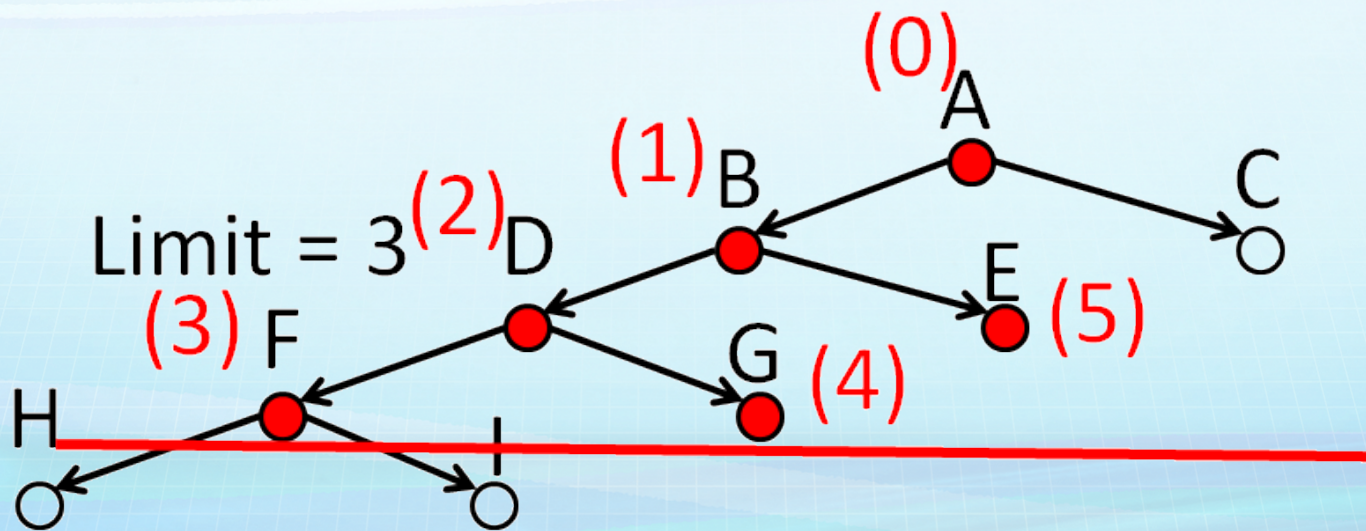
系統的探索

◆ 反復深化深さ優先探索



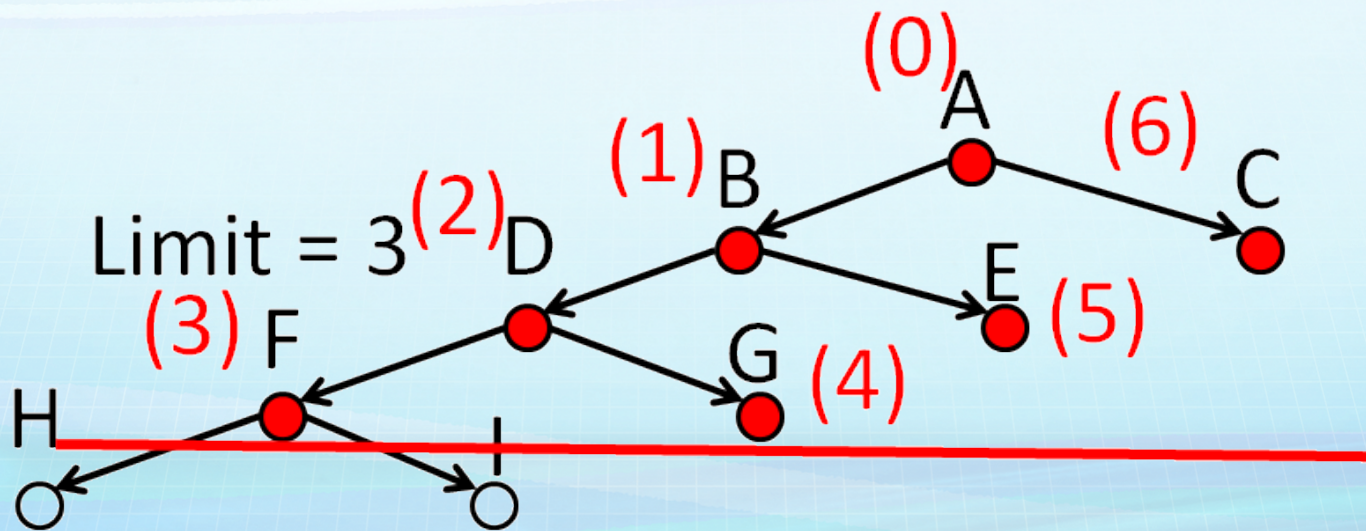
系統的探索

◆ 反復深化深さ優先探索



系統的探索

◆ 反復深化深さ優先探索



發見的探索

発見的探索

◆発見的（ヒューリスティック）探索

- 問題に関する知識を利用
- 系統的探索では現実的に解を発見できない場合
- 問題に関する知識は経験的知識を含む
 - 必ずしも最適解に導くとは限らない
 - 一般には最適解が得られる保証はない

発見的探索

◆山登り法

- 現在の状態から，より評価値の高い状態へ探索を進める
- アルゴリズムは単純
- 開始点から目標までの経路の各点において，子の点のうち，経路上の点の評価値が最も良くなければ，最適解には達しない

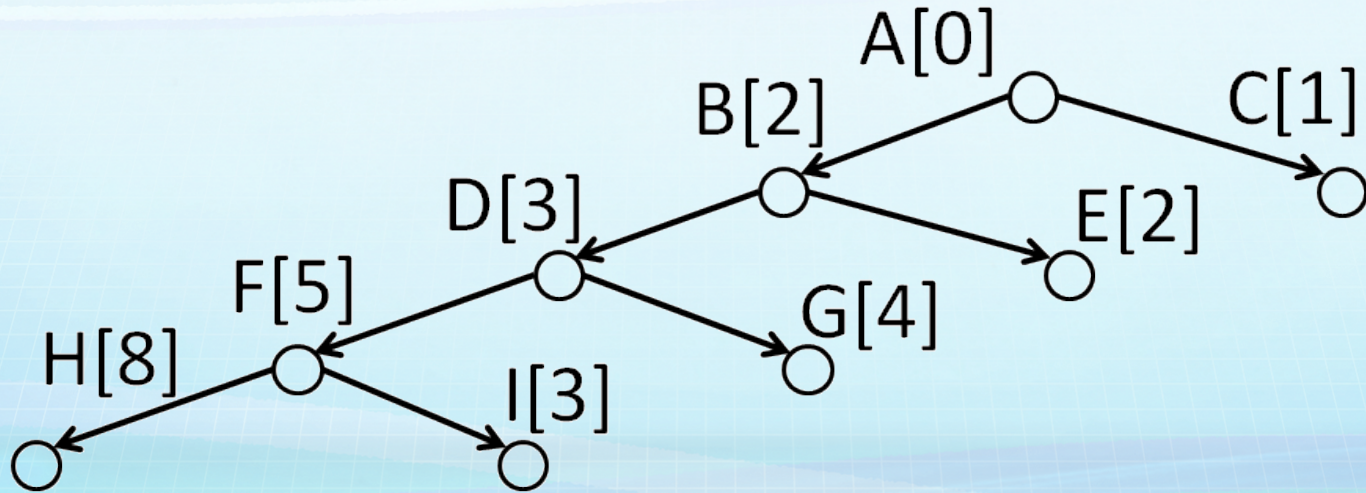
発見的探索

◆山登り法

- (0) $f(n)$: 点 n の評価関数 (ヒューリスティック関数)
- (1) 開始点を探索対象の点 n とする
- (2) もし, 点 n に適用可能な作用素がなければ, 点 n を解として出力して, 探索を終了
- (3) 点 n の子の点のうち, 評価値が $f(n)$ より良い点が存在しなければ, 点 n を解として出力して, 探索を終了. 評価値が $f(n)$ より良い点が存在すれば, 評価値が最も良い点を n として, (2)へ

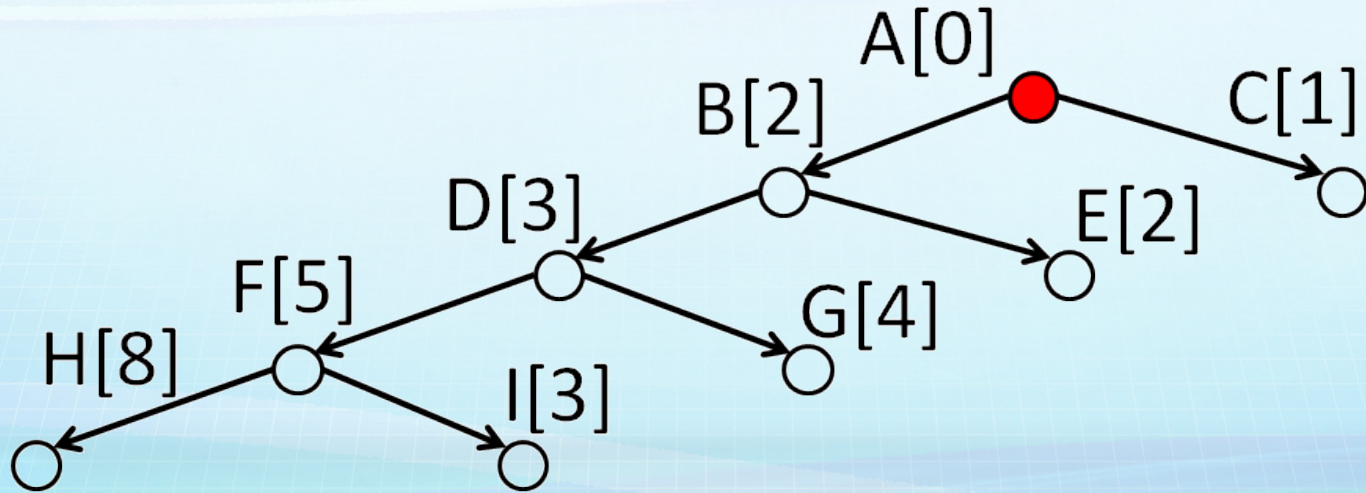
発見の探索

◆山登り法



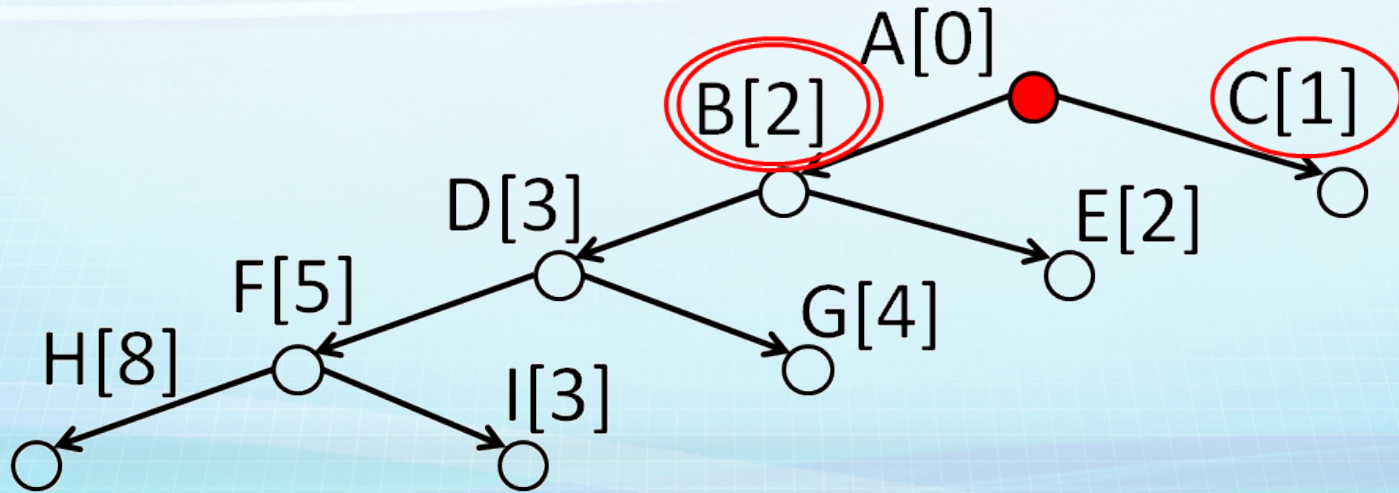
発見の探索

◆山登り法



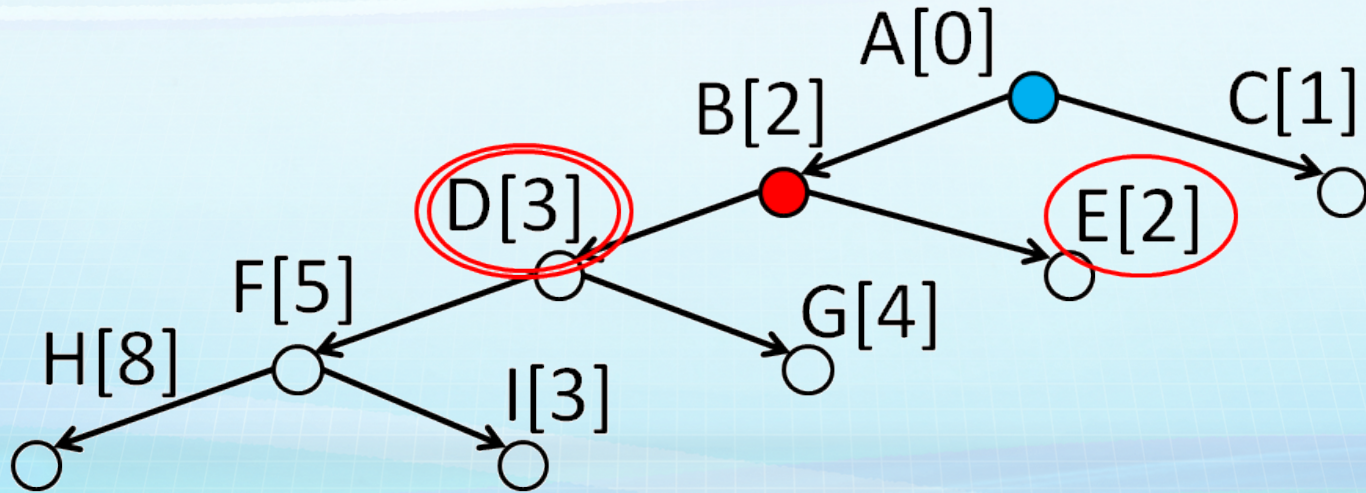
発見の探索

◆山登り法



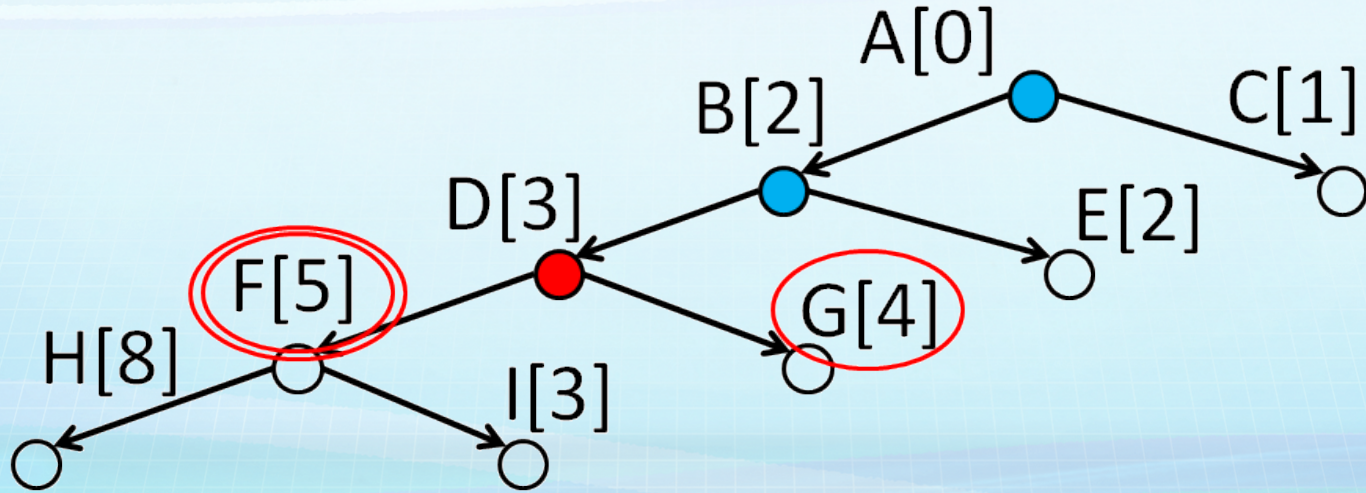
発見の探索

◆山登り法



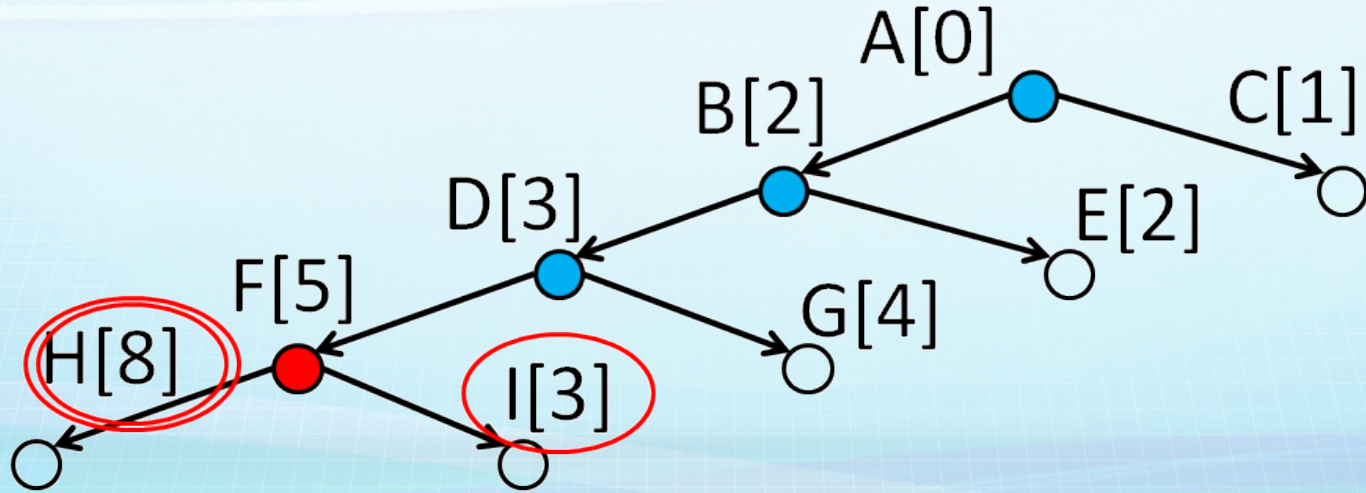
発見の探索

◆山登り法



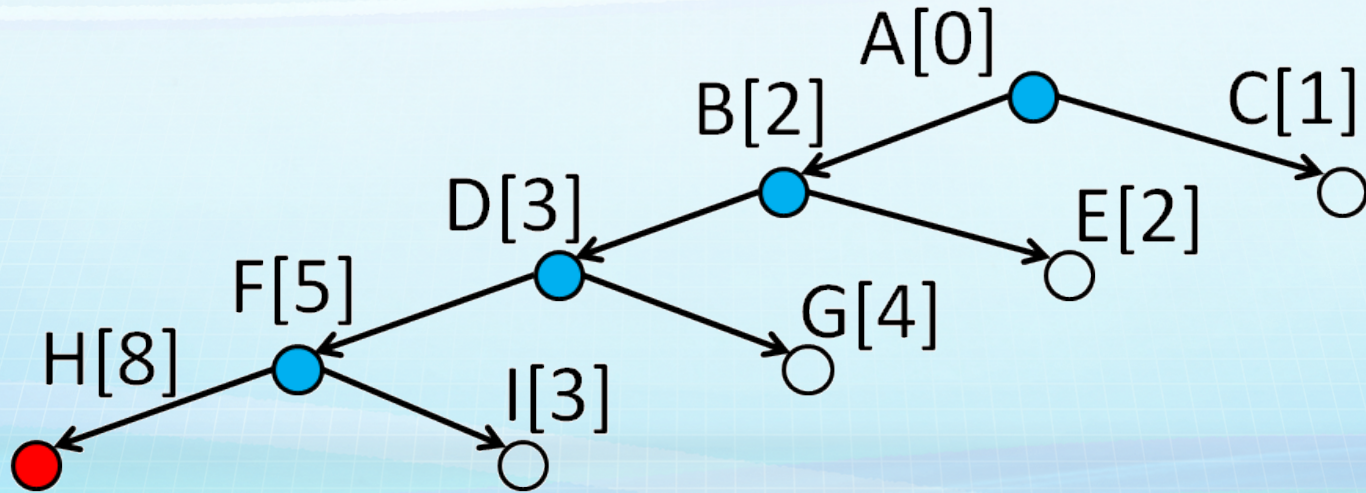
発見の探索

◆山登り法



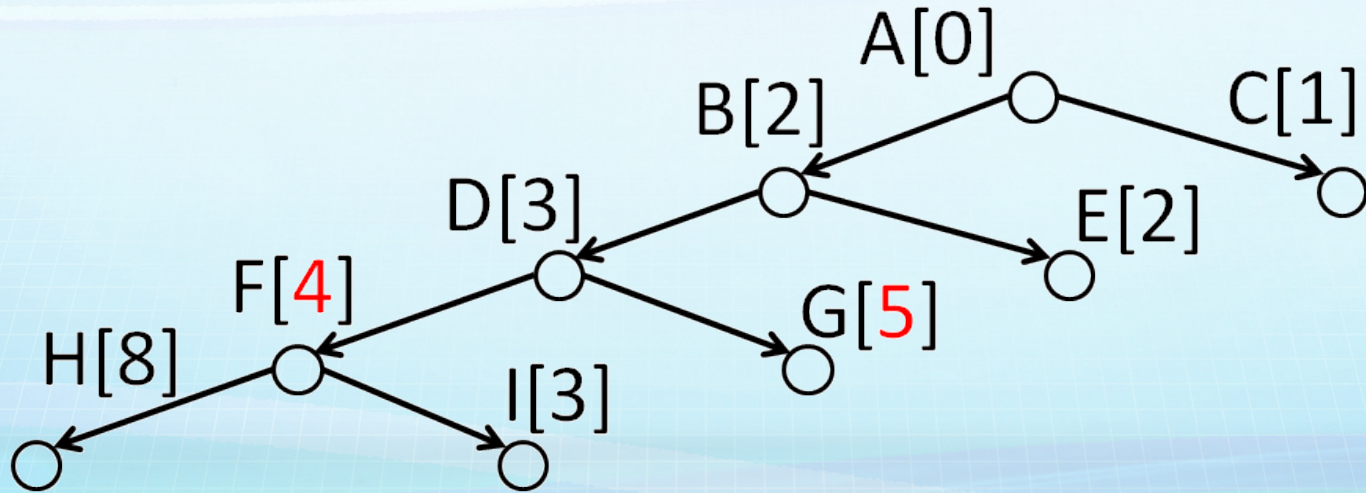
発見の探索

◆山登り法



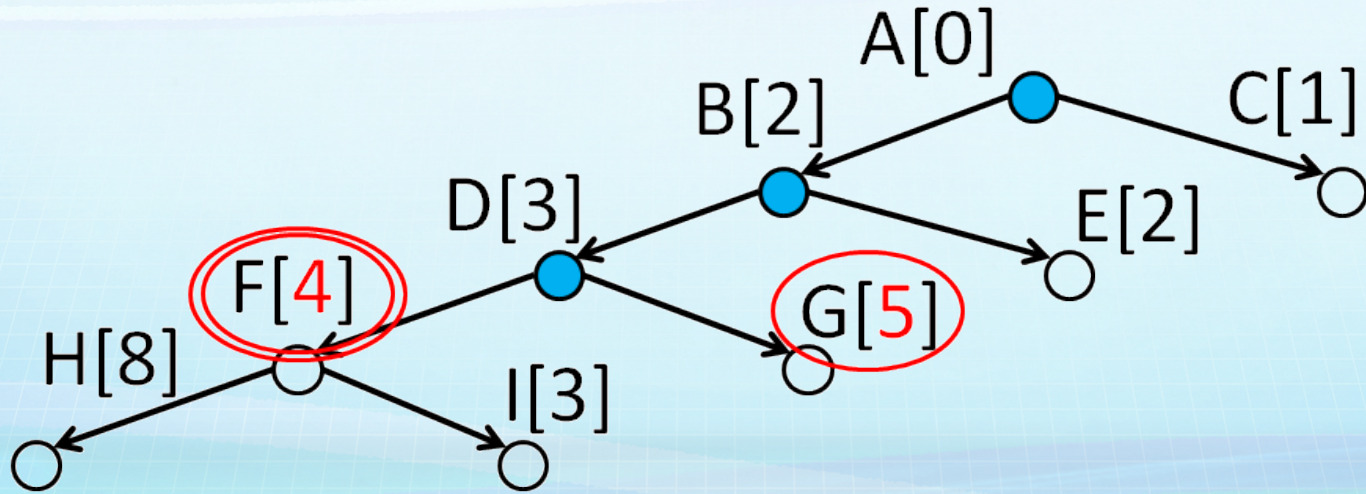
発見の探索

◆山登り法



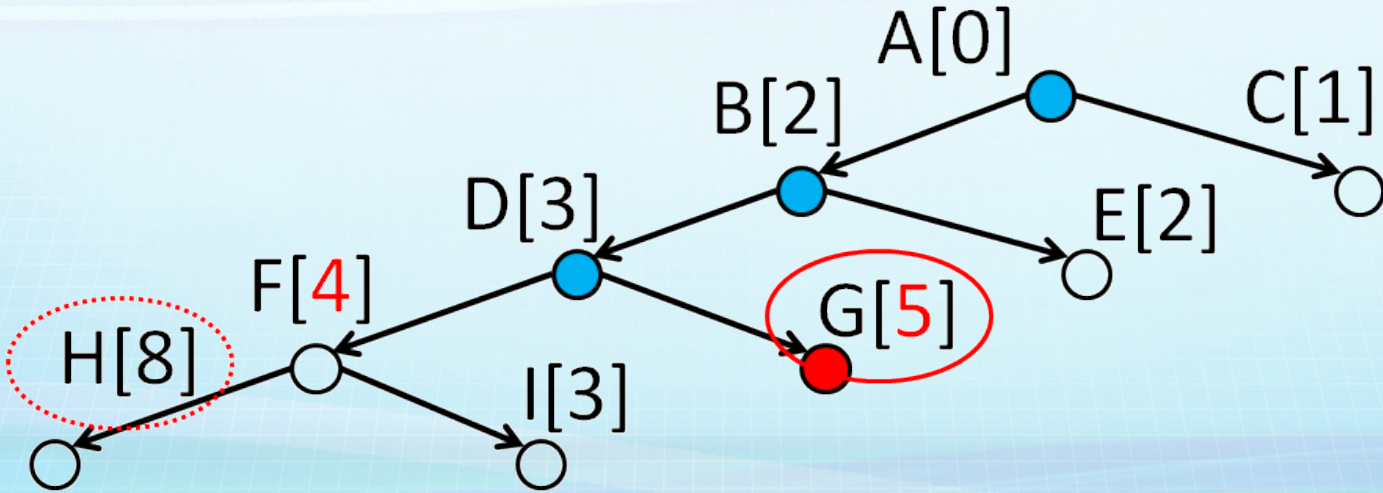
発見の探索

◆山登り法



発見の探索

◆山登り法



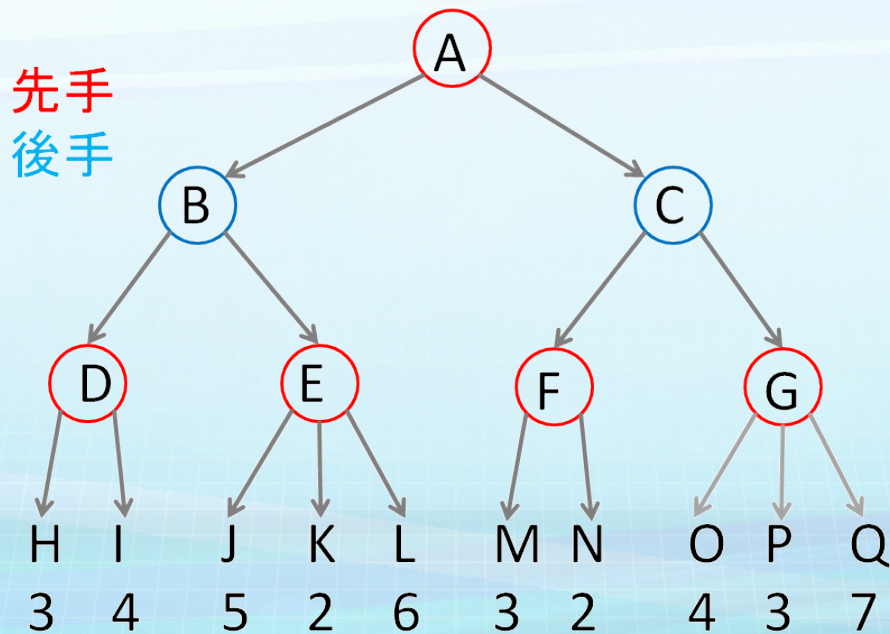
ゲームの木の探索

ゲームの木の探索

- チェス, 将棋, 三目並べ…
- 展開型ゲーム
 - ゲームの木で表現可能
 - 先読み推論で原理的には解ける
- 二人零和ゲーム … ミニマックス法で探索

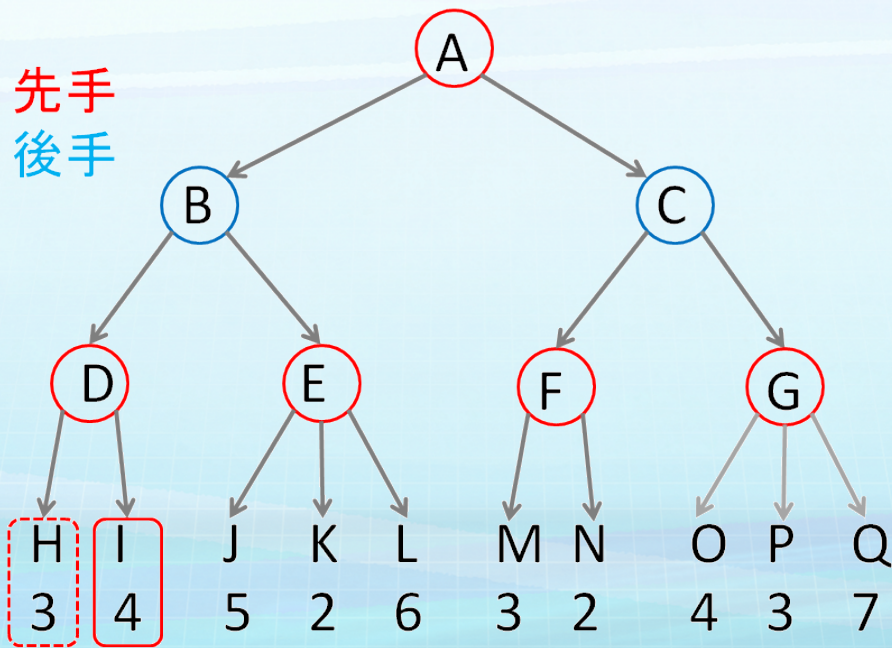
ゲームの木の探索

◆ミニマックス法



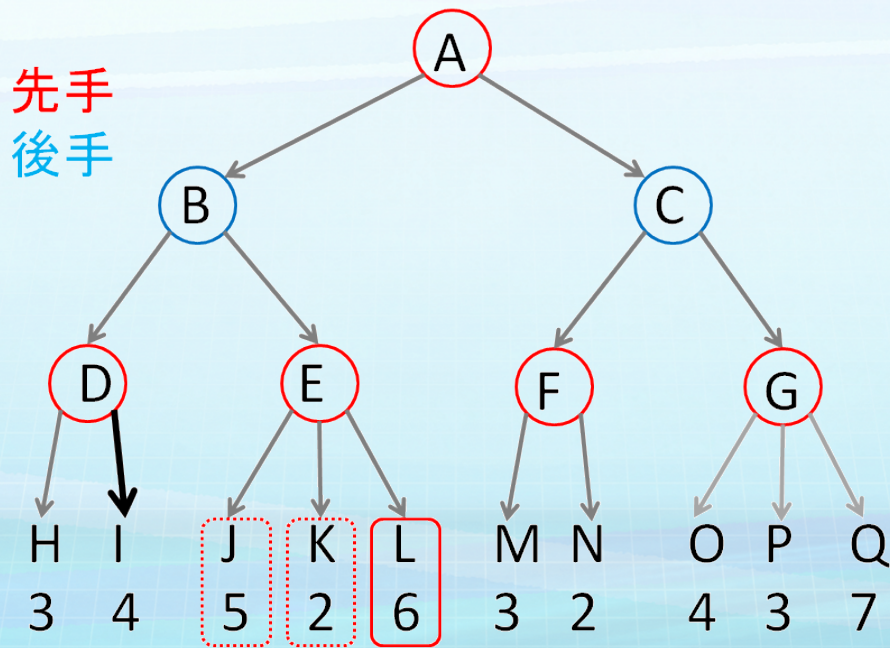
ゲームの木の探索

◆ミニマックス法



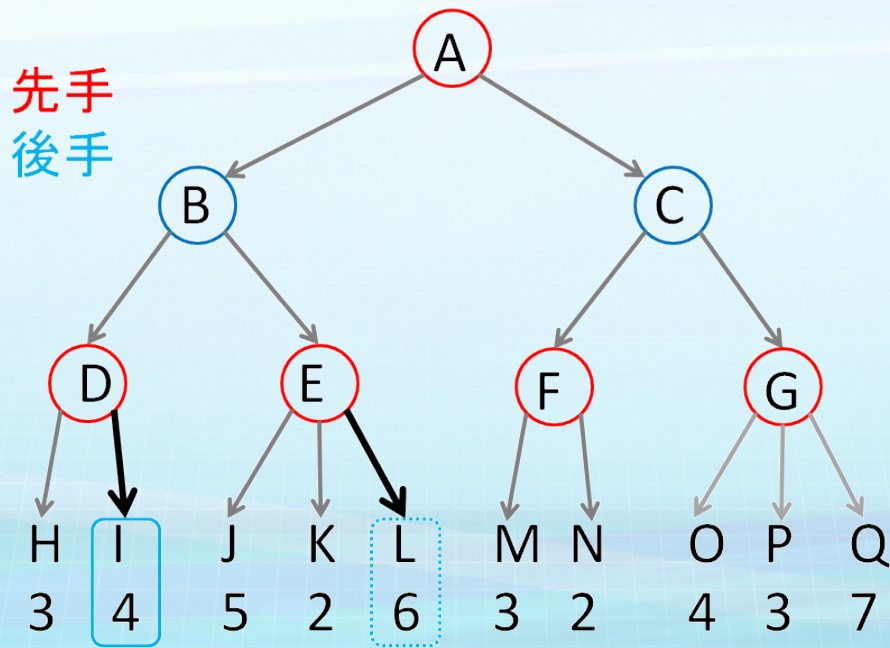
ゲームの木の探索

◆ミニマックス法



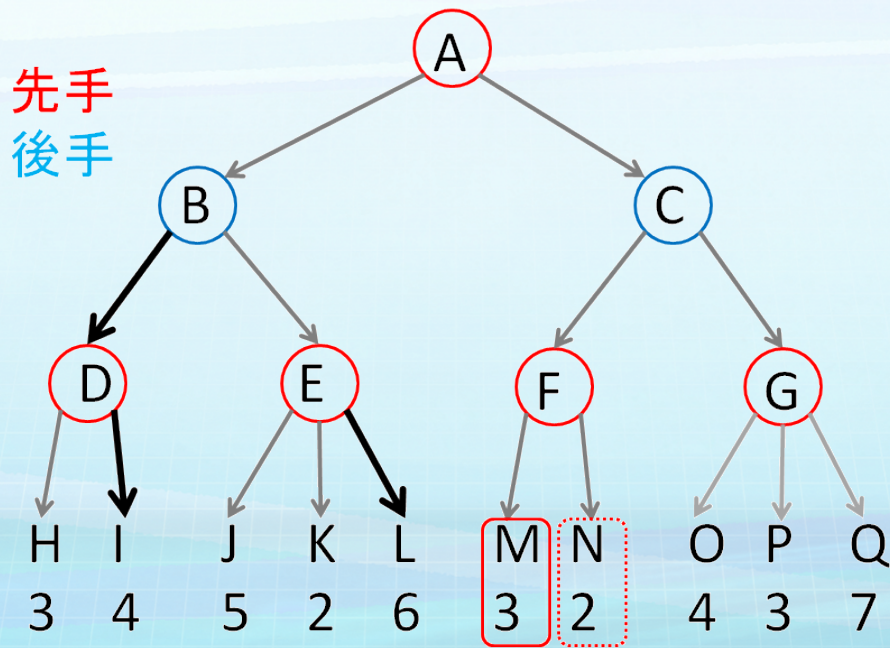
ゲームの木の探索

◆ミニマックス法



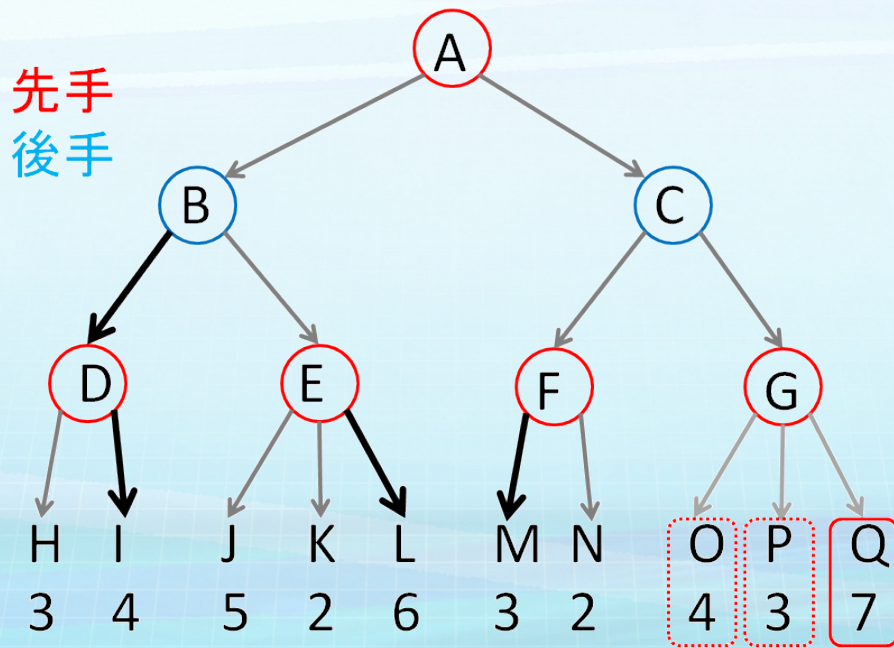
ゲームの木の探索

◆ミニマックス法



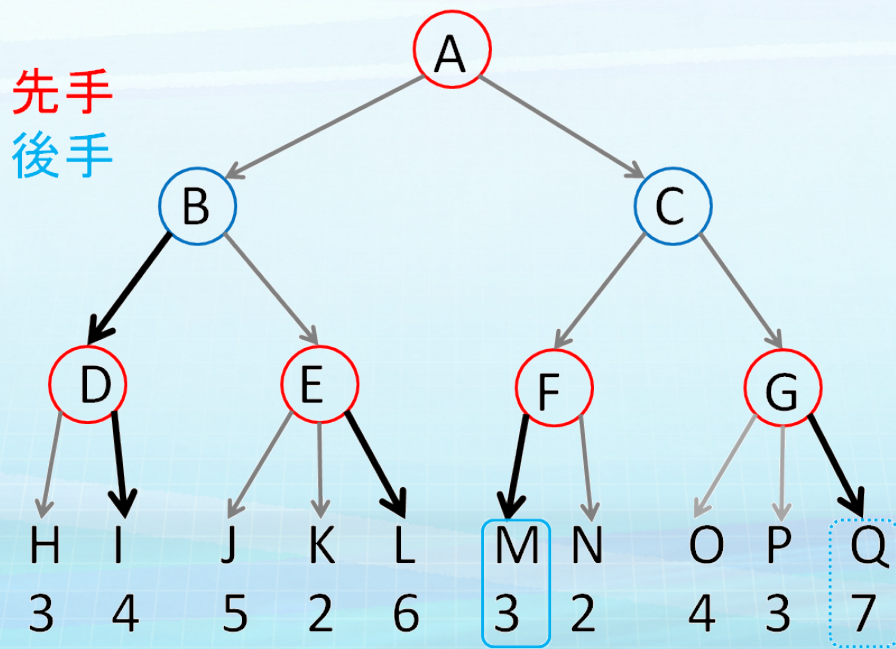
ゲームの木の探索

◆ミニマックス法



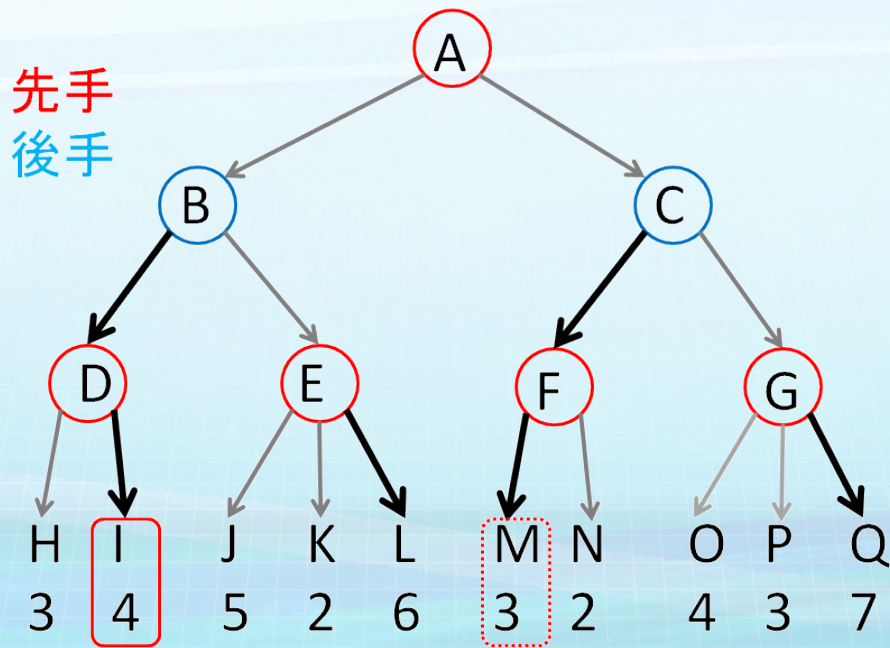
ゲームの木の探索

◆ミニマックス法



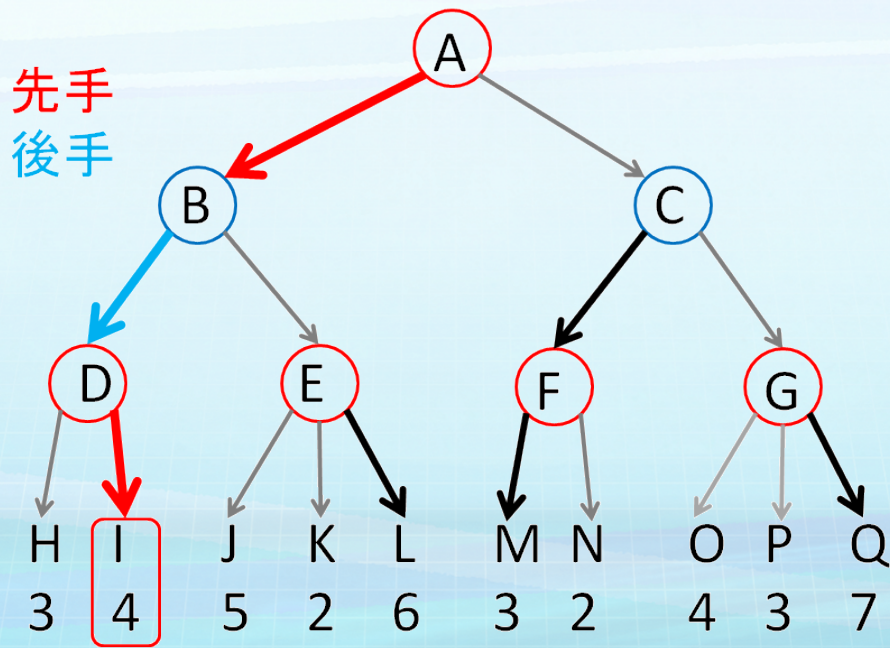
ゲームの木の探索

◆ミニマックス法



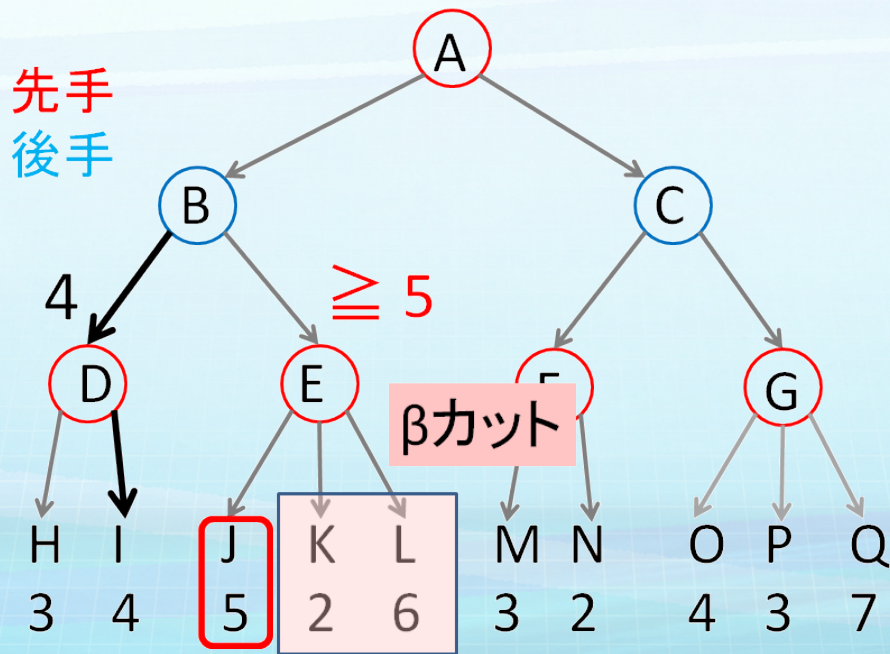
ゲームの木の探索

◆ミニマックス法



ゲームの木の探索

◆ $\alpha\beta$ 法



ゲームの木の探索

◆ $\alpha\beta$ 法

