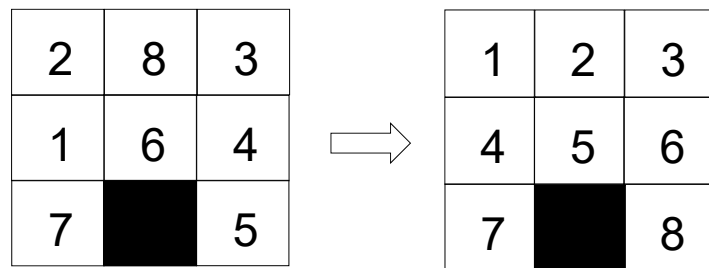


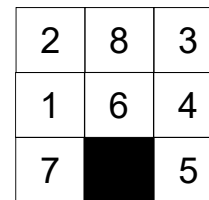
8パズル



初期状態

目標状態

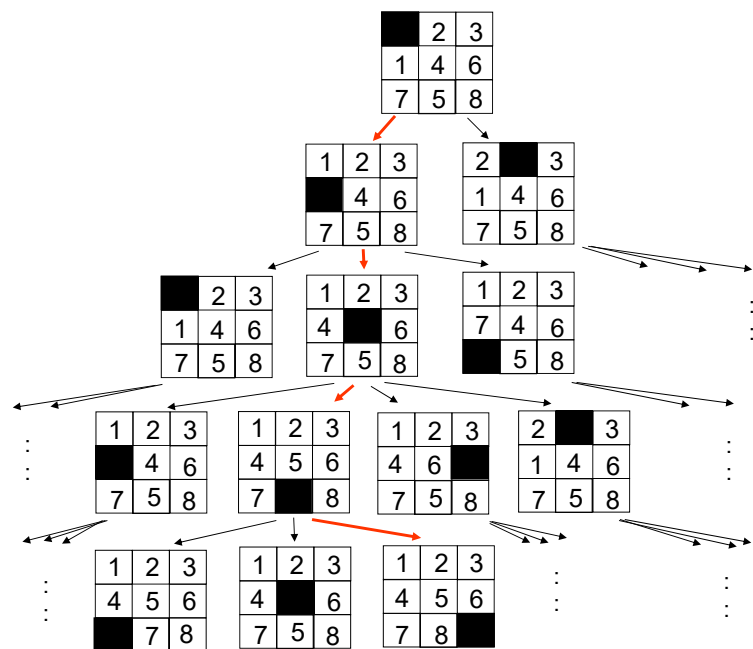
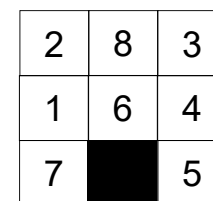
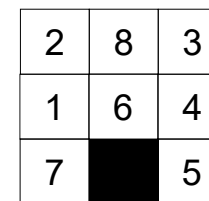
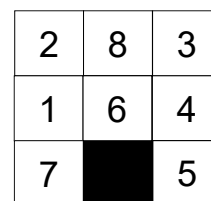
8パズル



up

right

left

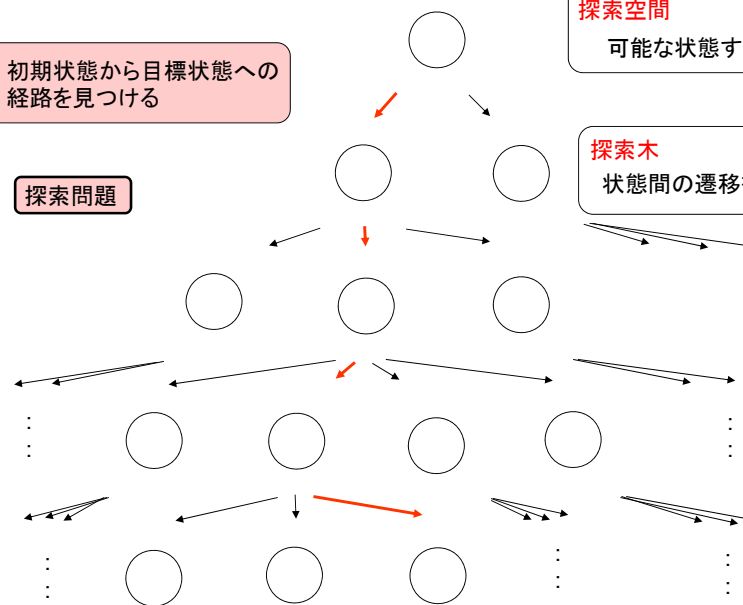


初期状態から目標状態への
経路を見つける

探索空間
可能な状態すべての集合

探索問題

探索木
状態間の遷移を表す木構造



探索の戦略(1)

- しらみつぶし探索(exhaustive search) - ブラインド探索(blind search), カマかせ探索(brute-force search)
 - 縦型探索(深さ優先探索)
 - 展開順はLIFO, スタック, 実装が簡単,
 - 計算量・メモリ量とも良,
 - 解の位置や木の深さによって解が発見できない場合あり
 - 横型探索(幅優先探索)
 - 展開順はFIFO, キュー,
 - 解の深さが探索木の深さより小さい時に効率良
 - 反復深化探索(iterative deepening)
 - 縦型探索を打ち切る深さを徐々に深くしたもの
 - 反復幅拡張探索

探索の戦略(2)

- 評価関数を利用した探索
 - 評価関数 $f(n) = g(n) + h(n)$
 - $f(n)$: ノード n の評価値, スタートからそのノードを経てゴールに行くコスト
 - $g(n)$: スタートからノード n までの評価値
 - $h(n)$: ノード n からゴールまでの評価値
 - $h'(n)$:
 - ノード n からゴールまでの予測値
 - $h(n)$ のかわりに使用,
 - ヒューリスティック関数(heuristic function)

最適解を得るための戦略

- 分岐(枝)限定法(branch and bound)}
 - non-heuristic の評価関数 $g(n)$ を使用,
 - 出発点からのコスト最小の節点から探索,
 - 解があれば最適解の発見を保証

ヒューリスティック探索(1) (heuristic search)

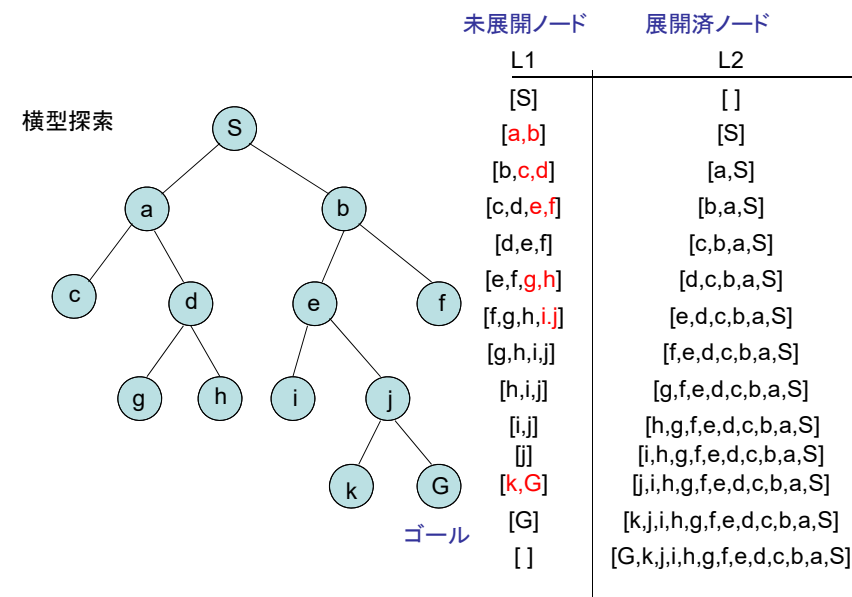
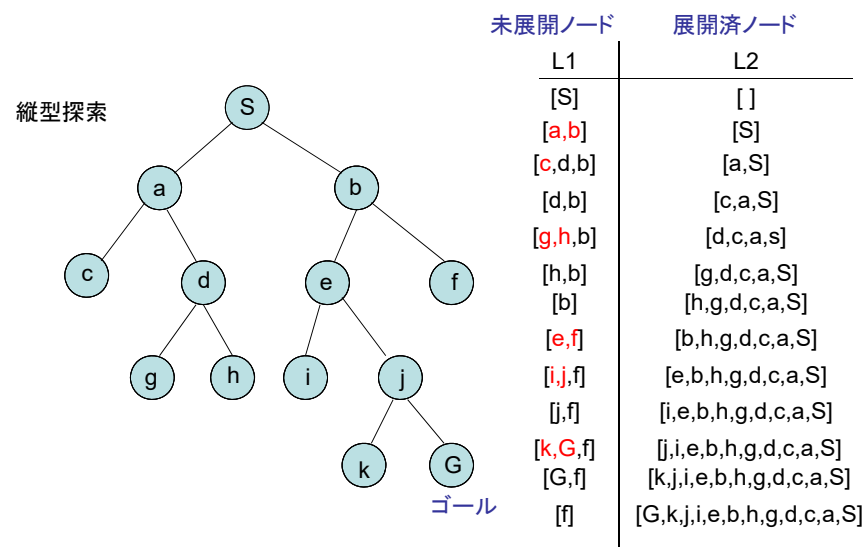
- 山登り法(hill-climbing)
 - 目標点までのコスト $h'(n)$ 最小の節点から探索
 - 現節点に接続する節点の集合から次に展開する節点を選択
 - 多峰の場合に局所的最小値(local minima)の問題あり

ヒューリスティック探索(2) (heuristic search)

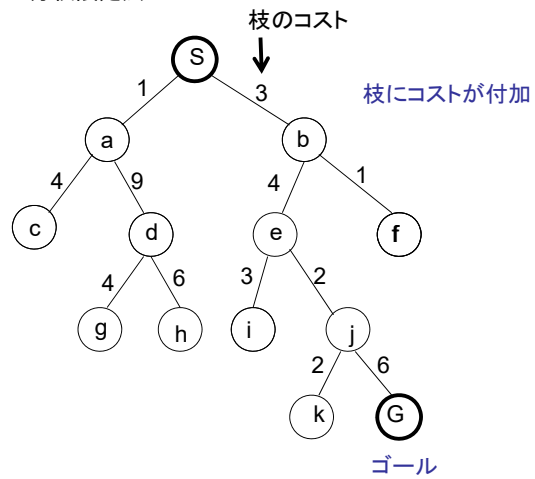
- 最良優先(best-first)
 - 目標点までのコスト $h'(n)$ 最小の節点から探索
 - その時点までに得られたすべての未展開節点から次に展開する節点を選択
 - 効率はよいがメモリ量大
 - 解があれば探索は成功(最適解の保証なし)

ヒューリスティック探索(3) (heuristic search)

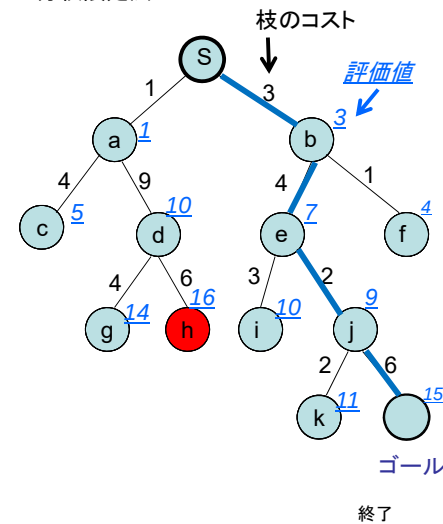
- A探索
 - 出発点からのコストと目標点までのコストの和 $f(n)$ が最小の節点から探索,
 - 評価関数が $h'(n) \leq h(n)$ の場合には最適解発見を保証(この時に **A*探索** という)
- その他
 - 焼きなまし法(simulated annealing)
 - タブー探索



分枝限定法



分枝限定法



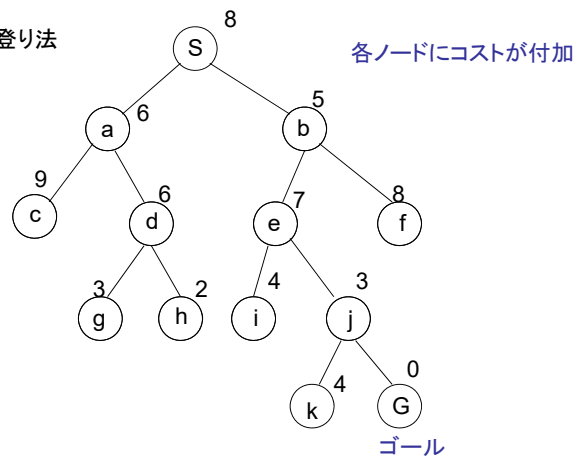
未展開ノード

L1

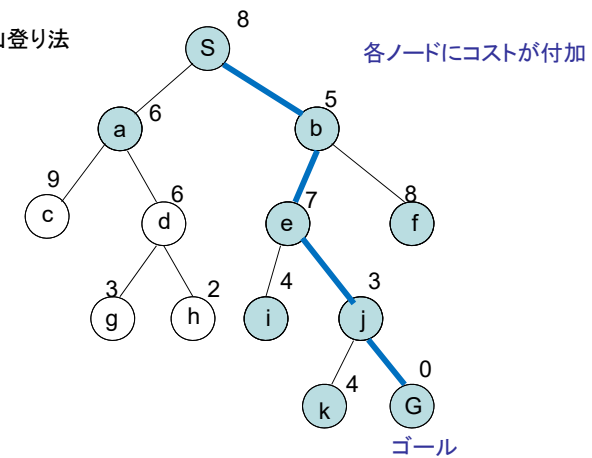
[S(0)] ← ノード(評価値)
 [a(1), b(3)]
 [b(3), c(5), d(10)]
 [f(4), c(5), e(7), d(10)]
 [c(5), e(7), d(10)]
 [e(7), d(10)]
 [i(9), d(10), i(10)]
 [d(10), i(10), k(11), G(15)]
 [i(10), k(11), g(14), G(15), h(16)]
 [k(11), g(14), G(15), h(16)]
 [g(14), G(15), h(16)]
 [G(15), h(16)]
 [h(16)]

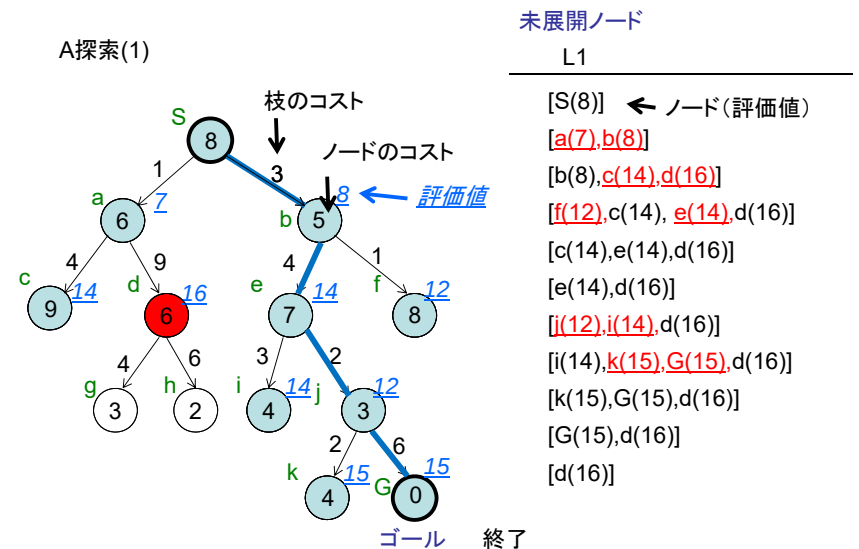
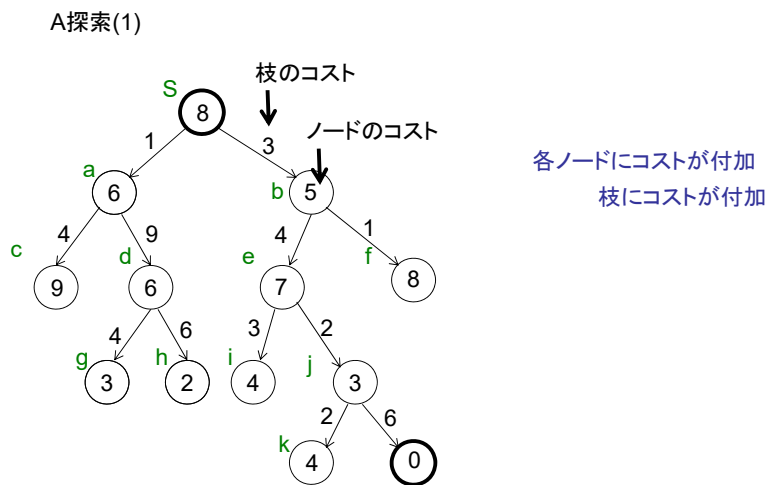
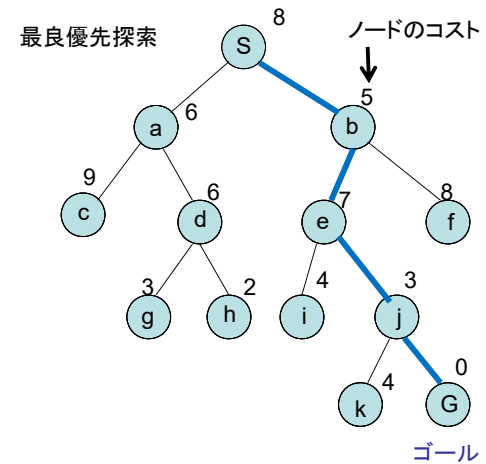
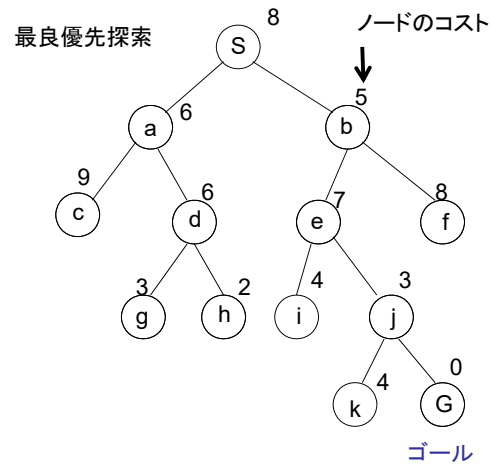
展開済ノード L2 [G(15), g(14), k(11), i(10), d(10), j(9), e(7), c(5), f(4), b(3), a(1), S(0)]

山登り法



山登り法





[S(8)] ← ノード(評価値)
[\[a\(7\), b\(8\)\]](#)
 [b(8), [c\(14\)](#), d(16)]
 [f(12), c(14), [e\(14\)](#), d(16)]
 [c(14), e(14), d(16)]
 [e(14), d(16)]
 [[i\(12\)](#), [i\(14\)](#), d(16)]
 [i(14), [k\(15\)](#), [G\(15\)](#), d(16)]
 [k(15), G(15), d(16)]
 [G(15), d(16)]
 [d(16)]

下線のついてるのは新しく生成されたノード

A探索(2)

