Iterative Deepening Search

$O(b^d)$ $O(bd)$

function ITERATIVE-DEEPENING-SEARCH(*problem*) **returns** a solution, or failure
  for *depth* = 0 **to** ∞ **do**
    result ← DEPTH-LIMITED-SEARCH(*problem*, *depth*)
    if *result* ≠ cutoff **then return** *result*

Figure 3.18 The iterative deepening search algorithm, which repeatedly applies depth-limited search with increasing limits. It terminates when a solution is found or if the depth-limited search returns *failure*, meaning that no solution exists.

Iterative Deepening Search

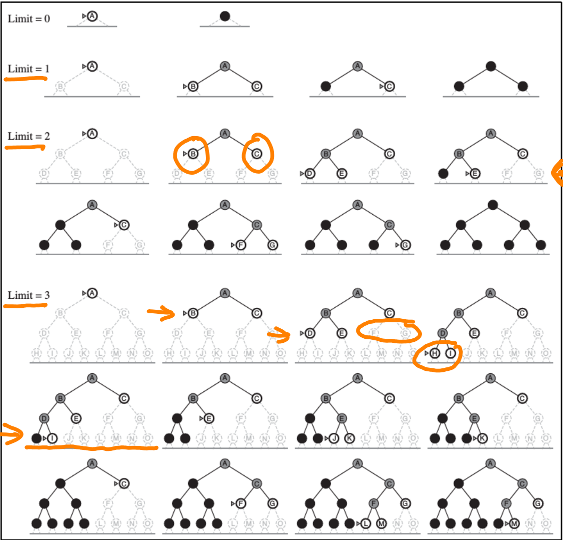


Figure 3.19 Four iterations of iterative deepening search on a binary tree.

d

$(d-1)$

b

bd

1

b^1

b^2

b^d

Iterative Deepening Search

Time complexity:

$$N(\text{IDS}) = (d)b + (d-1)b^2 + \cdots + (1)b^d, \quad = O(b^d)$$

if $b = 10$ and $d = 5$, the numbers are

$$N(\text{IDS}) = 50 + 400 + 3,000 + 20,000 + 100,000 = 123,450$$

$$N(\text{BFS}) = 10 + 100 + 1,000 + 10,000 + 100,000 = 111,110$$

Iterative Deepening Search

Time complexity:

$$N(\text{IDS}) = (d)b + (d-1)b^2 + \cdots + (1)b^d \ ,$$

if $b = 10$ and $d = 5$, the numbers are

$$\begin{aligned} N(\text{IDS}) &= 50 + 400 + 3,000 + 20,000 + 100,000 = 123,450 \\ N(\text{BFS}) &= 10 + 100 + 1,000 + 10,000 + 100,000 = 111,110 \end{aligned}$$

- Complete and optimal?

Iterative Deepening Search

Time complexity:

$$N(\text{IDS}) = (d)b + (d-1)b^2 + \cdots + (1)b^d \,,$$

if $b = 10$ and $d = 5$, the numbers are

$$\begin{aligned} N(\text{IDS}) &= 50 + 400 + 3,000 + 20,000 + 100,000 = 123,450 \\ N(\text{BFS}) &= 10 + 100 + 1,000 + 10,000 + 100,000 = 111,110 \end{aligned}$$

- Complete and optimal?
- Space complexity?

$o(b^d)$

Iterative Deepening Search

Time complexity:

$$N(\mathrm{IDS}) = (d)b + (d-1)b^2 + \cdots + (1)b^d \ ,$$

if $b = 10$ and $d = 5$, the numbers are

$$\begin{aligned} N(\mathrm{IDS}) &= 50 + 400 + 3,000 + 20,000 + 100,000 = 123,450 \\ N(\mathrm{BFS}) &= 10 + 100 + 1,000 + 10,000 + 100,000 = 111,110 \end{aligned}$$

- Complete and optimal?
- Space complexity? $O(bd)$

Bidirectional Search

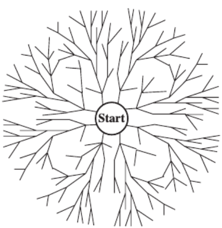


Figure 3.20 A schematic view of a bidirectional search that is about to succeed when a branch from the start node meets a branch from the goal node.

Bidirectional Search

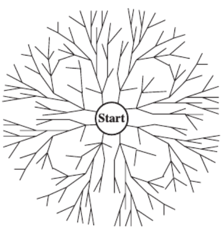


Figure 3.20 A schematic view of a bidirectional search that is about to succeed when a branch from the start node meets a branch from the goal node.

- Can we do bidirectional search for 8 puzzle problem?

Bidirectional Search

- Will bidirectional search be useful? Why?

$d = 10$

$b = 2$

$\underbrace{b^d}$ $\quad 2^{10} = \underbrace{1024}$

$2 \cdot \; b^{d/2} \quad 2^5 = 32$

$\underline{\underline{64}}$

Bidirectional Search

- Will bidirectional search be useful? Why?
- Can bidirectional search be used for all problems?

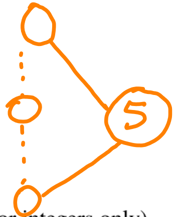
Bidirectional Search

- Will bidirectional search be useful? Why?
- Can bidirectional search be used for all problems?
- Knuth's problem

Bidirectional Search

$[5,6)$

- Will bidirectional search be useful? Why?
- Can bidirectional search be used for all problems?
- Knuth's problem



The problem definition:

- **States**: Positive numbers.
- **Initial state**: 4.
- **Actions**: Apply factorial, square root, or floor operation (factorial for integers only).
- **Transition model**: As given by the mathematical definitions of the operations.
- **Goal test**: State is the desired positive integer.

$\sqrt{\sqrt{\sqrt{\sqrt{5}}}}$

$$\left\lfloor \sqrt{\sqrt{\sqrt{\sqrt{\sqrt{(4!)!}}}}} \right\rfloor = \boxed{5}$$

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?						
Time						
Space						
Optimal?						

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ;

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes^a					
Time	$O(b^d)$					
Space	$O(b^d)$					
Optimal?	Yes^c					

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: ^{a} complete if b is finite; ^{b} complete if step costs $\geq \epsilon$ for positive ϵ;

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]				
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$				
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$				
Optimal?	Yes[c]	Yes				

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]	No			
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$			
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$			
Optimal?	Yes[c]	Yes	No			

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]	No	No		
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$		
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$		
Optimal?	Yes[c]	Yes	No	No		

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes^a	Yes^{a,b}	No	No	Yes^a	
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	
Optimal?	Yes^c	Yes	No	No	Yes^c	

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: ^{a} complete if b is finite; ^{b} complete if step costs $\geq \epsilon$ for positive ϵ; ^{c} optimal if step costs are all identical; ^{d} if both directions use breadth-first search.

Uninformed Search Strategies

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes^a	Yes^{a,b}	No	No	Yes^a	Yes^{a,d}
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes^c	Yes	No	No	Yes^c	Yes^{c,d}

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: ^{a} complete if b is finite; ^{b} complete if step costs $\geq \epsilon$ for positive ϵ; ^{c} optimal if step costs are all identical; ^{d} if both directions use breadth-first search.

Informed (Heuristic) Search Strategies

- Evaluation function $f(n)$
- Heuristic function $h(n)$

$g(n)$

$f(n)$

A^* search

$f(n) = g(n) + h(n)$

Informed (Heuristic) Search Strategies

$f(n) = g(n) + h(n)$

$g(s) + h(s)$

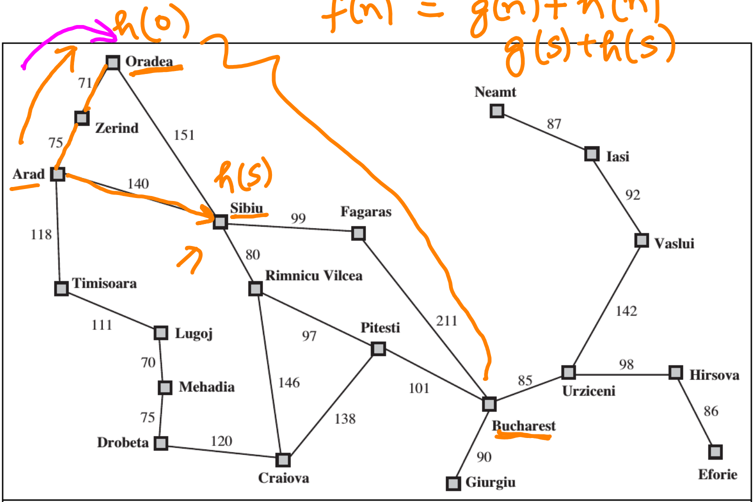


Figure 3.2 A simplified road map of part of Romania.

Straight Line Distances

h(

Arad	366	**Mehadia**	241
Bucharest	0	**Neamt**	234
Craiova	160	**Oradea**	380
Drobeta	242	**Pitesti**	100
Eforie	161	**Rimnicu Vilcea**	193
Fagaras	176	**Sibiu**	253
Giurgiu	77	**Timisoara**	329
Hirsova	151	**Urziceni**	80
Iasi	226	**Vaslui**	199
Lugoj	244	**Zerind**	374

Figure 3.22 Values of h_{SLD}—straight-line distances to Bucharest.

Greedy Best-first Search

- Evaluation function $\underbrace{f(n)} = h(n)$

Greedy Best-first Search

- Evaluation function $f(n) = h(n)$

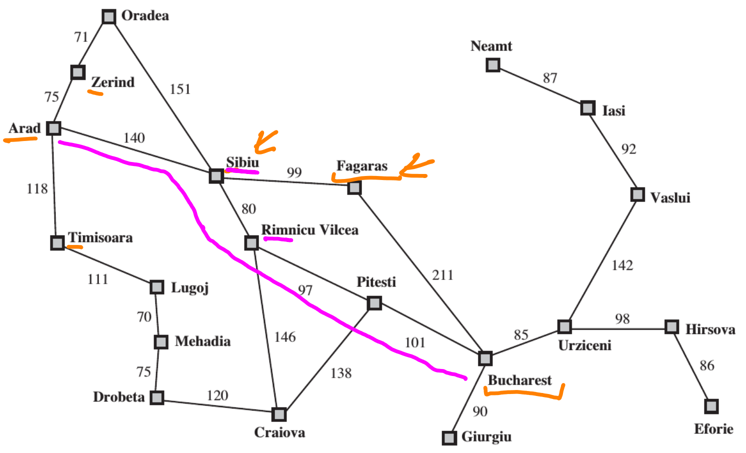


Figure 3.2 A simplified road map of part of Romania.