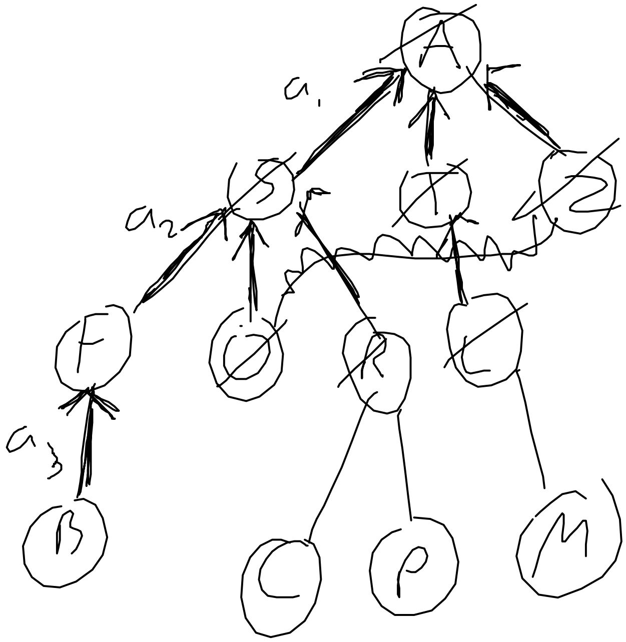
BFS

Wednesday, September 7, 2016 8:56 AM

* Successor fn → alpha order

Open: ~~A~~ ~~S~~ ~~T~~ ~~Z~~ ~~F~~ ~~O~~ ~~R~~ ~~L~~ ~~O~~ B C P M

Closed: A S T Z F O R ~~L~~



Solution: A → S

S → F

F → B

Recreate the path

state keeps a pointer

to its parent

& which action was used

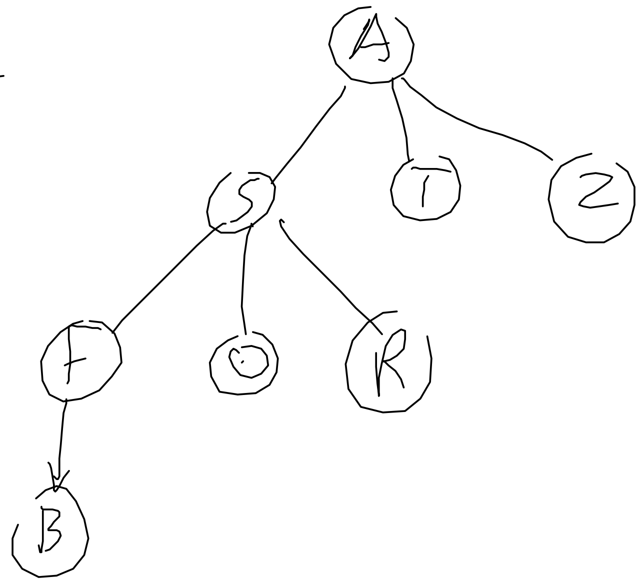
DFS

Wednesday, September 7, 2016 8:56 AM

Successor in alpha order

open: A ~~B~~ ~~F~~ O R ~~S~~ T Z ~~A~~

closed: A S F B

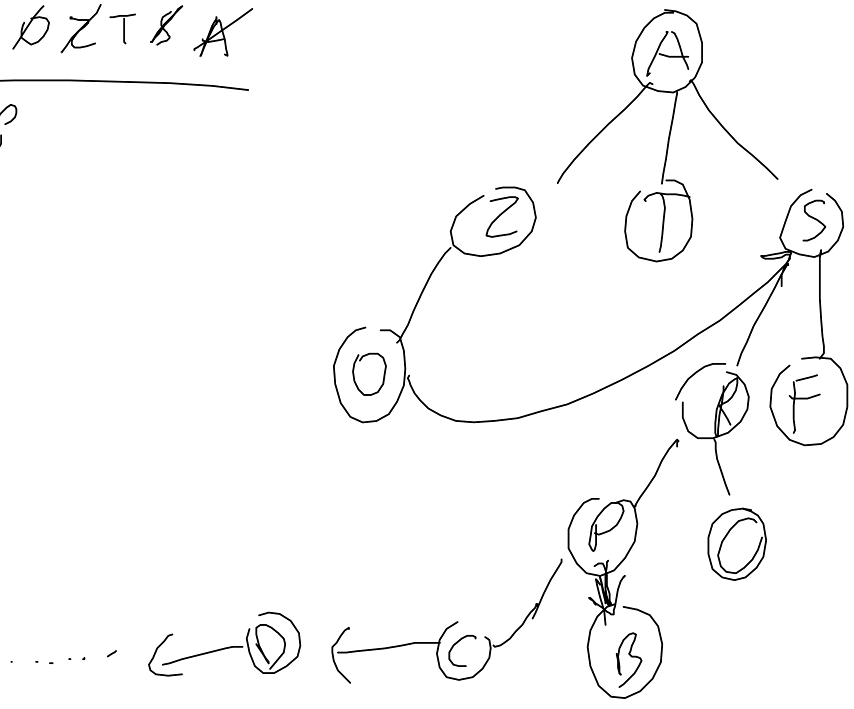


Solution: A → S
S → F
F → B

Successors in reverse alpha

open: C B ~~P~~ ~~C~~ ~~R~~ F S ~~O~~ ~~Z~~ T ~~S~~ ~~A~~

closed: A Z O S R P



Search 3

Wednesday, ~~September~~ 7, 2016 8:56 AM

BFS

Completeness: Yes - search at each depth

Optimality: Yes

Time: branching factor - avg # of actions per state

Exponential $O(\underbrace{b}_{\text{branching factor}}{}^{n})$ solution depth

Space: → All nodes generated Kept in memory

DFS

Completeness: Yes if state space is not infinite

No if ∞

Optimality: No

Time: Exponential - Avg. time complexity is lower than BFS

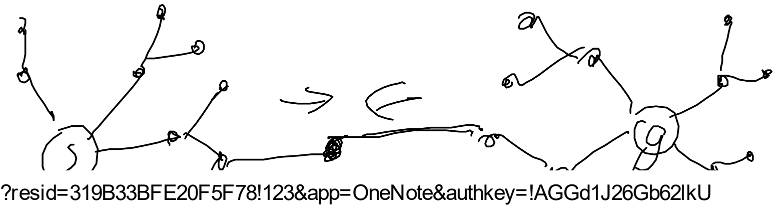
$O(b^{n})$ max depth of expansions

Space: Keep all nodes in memory

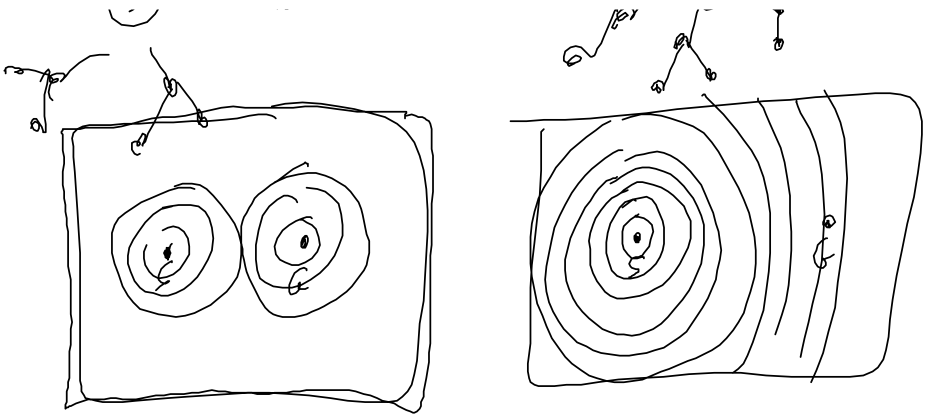
→ free-spaces - Keep the current branch in memory

Bi-Directional Search

~BFS but expand from init state & goal state

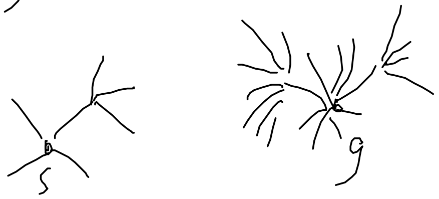


i In

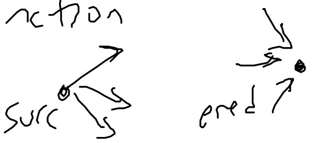


complete time $O(b^{n/2})$

- branching factor of going backwards is higher than the bf going forwards → bad idea



- have to have a predecessor function



Uniform Cost Search

Wednesday, September 7, 2016 9:11 AM

Actions w/ costs

Instead of shortest # of action → lowest cost solutions

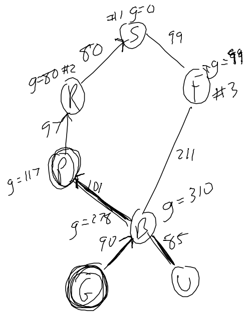
- optimal from cost perspective
- complete

g(n) - cost of state n from init state to n via the shortest known path

g(init) = 0

init: S

Goal: G



Action cost

Uniform cost search

g(n) - actual cost from init to state n ~~a~~ along the best known path.

Sort the open list with g(n)

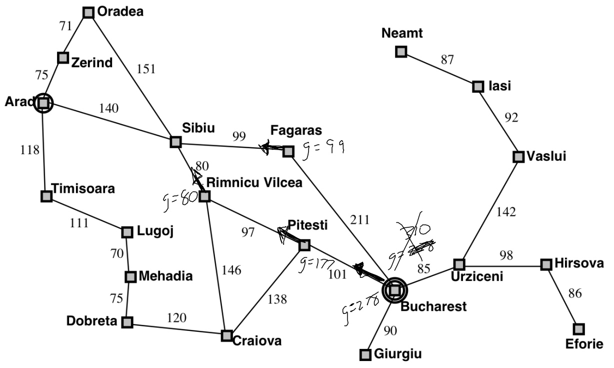
open list → priority Queue

Generic Search Algorithm

```
ops <— {…}
closed <— nil
open <— {initial-state}
current <— initial-state
WHILE (NOT isgoal(current) AND open ≠ NIL) DO:
    closed <— closed + {current}
    open <— open - {current} ⊕ ( successors(current, ops) - closed )
    current <— first(open)
END WHILE
IF isgoal(current) THEN report success!
ELSE report failure
```

insert sort based on g()
O(n)

Example: Romania



Chapter 3 6

Handling shortcuts

```
WHILE (NOT isgoal(current) AND open ≠ NIL) DO:
    closed <— closed + {current}    visit
    FOREACH n ∈ successors(current, ops) DO:
        IF n is not on open or closed THEN DO:
            compute g(n)
            Insert n into open (ranked on g())
        ELSE IF n is on open AND n is reached by a shorter path THEN DO:    smaller g for n
            n.parent <— current  ←
            update g(n)  ←    Delete + reinsert
            resort open  ←    O(n)
    END FOREACH
    current <— pop(open)
END WHILE
```

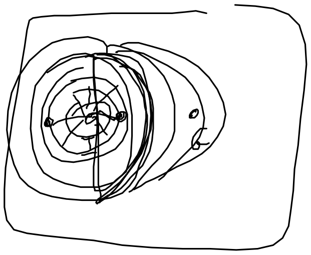
New

UCS is optimal & complete

→ but you cant jump straight to goal state
when successor function generates it.

Informed search

Friday, September 9, 2016 9:19 AM



Heuristic Search

intuition – programmer have some intuition that makes the problem easy(er) to solve

<u>Heuristic function</u> – captures intuition
focus on certain states above others

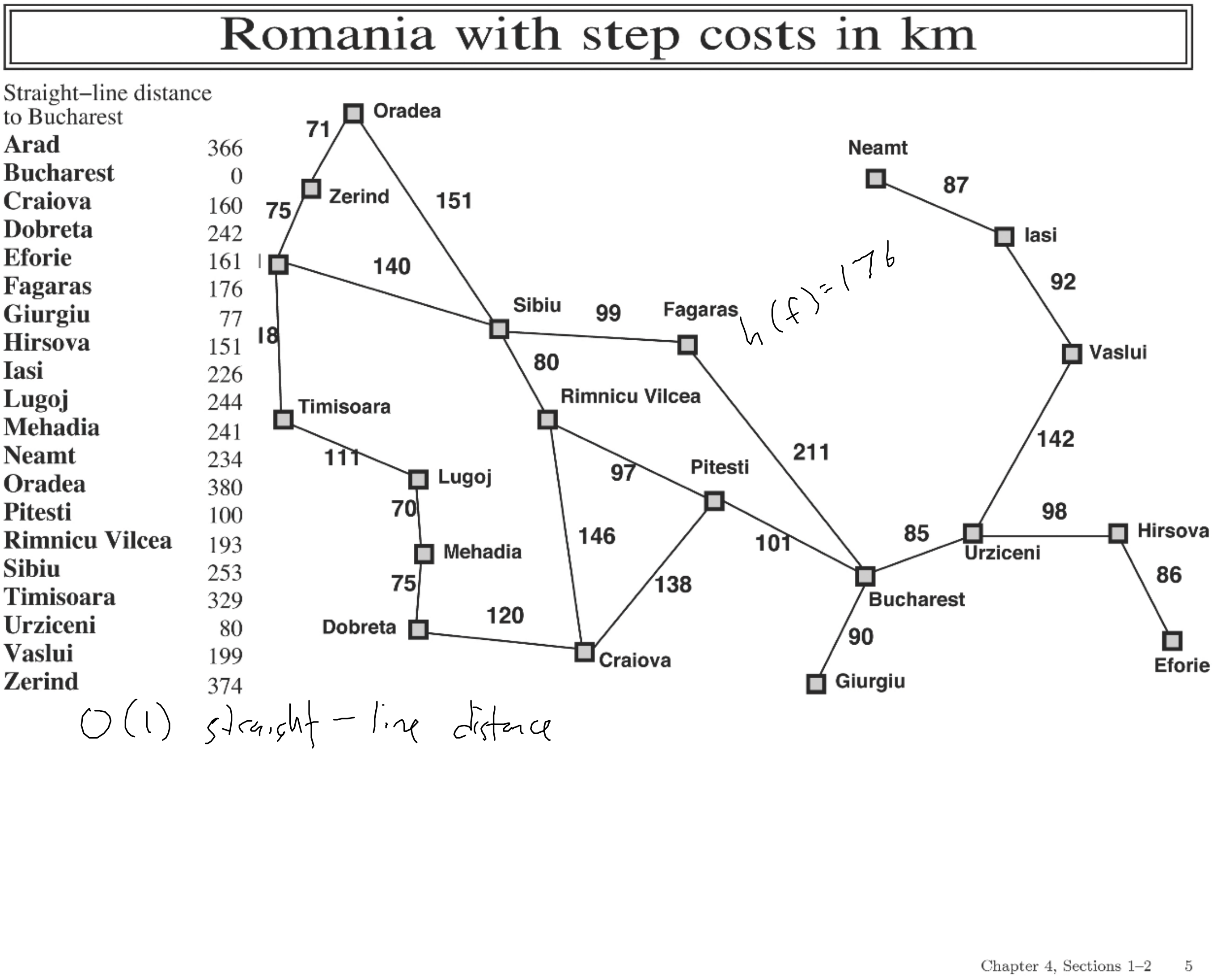
$h(n)$ h: state → number
↑ "Goodness" value

specifically: estimate of how close I am to goal.
(action cost or # of actions)

Romania Heuristic

Monday, September 12, 2016 9:25 AM

romania-sld-fixed2



Romania with step costs in km

Straight−line distance to Bucharest

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

O(1) straight-line distance

Chapter 4, Sections 1–2 5

Greedy Search

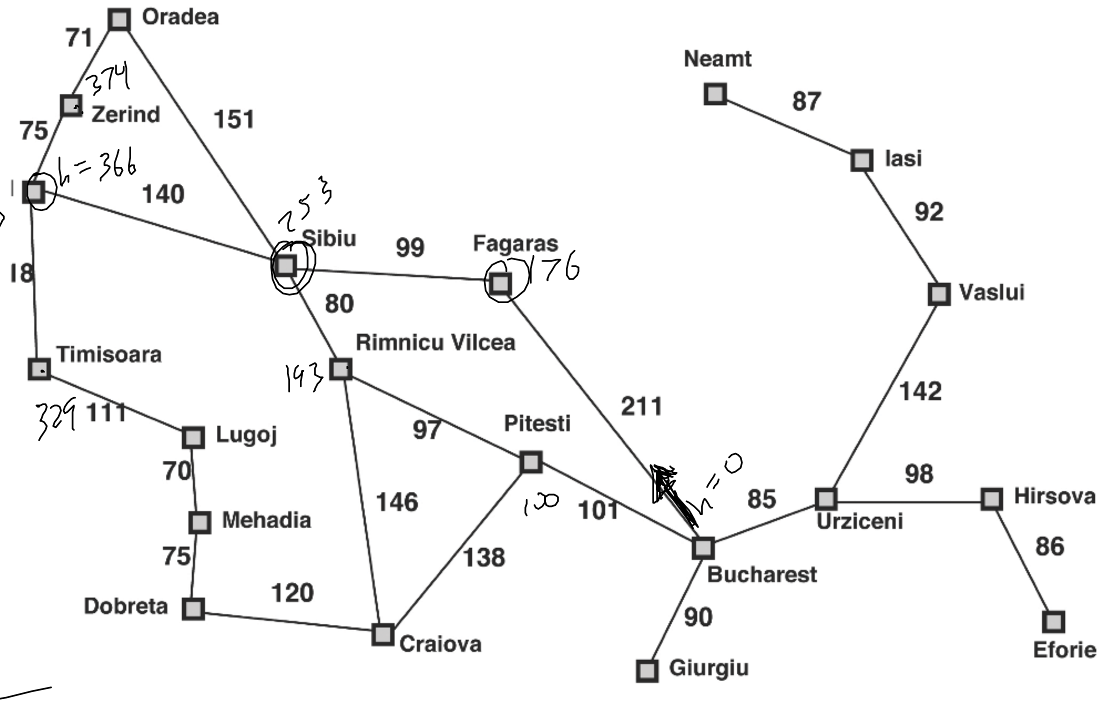
Monday, September 12, 2016 9:35 AM

open list is priority queue

sorted by h(n)

Romania with step costs in km

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



$$h() = \sqrt{(x_1 - x_2)^2 \ldots}$$

Chapter 4, Sections 1–2 5

Best first search

Friday, September 9, 2016 9:19 AM

Merge UCS & greedy search

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

Actual cost from init state to n by shortest known path

Be skeptical

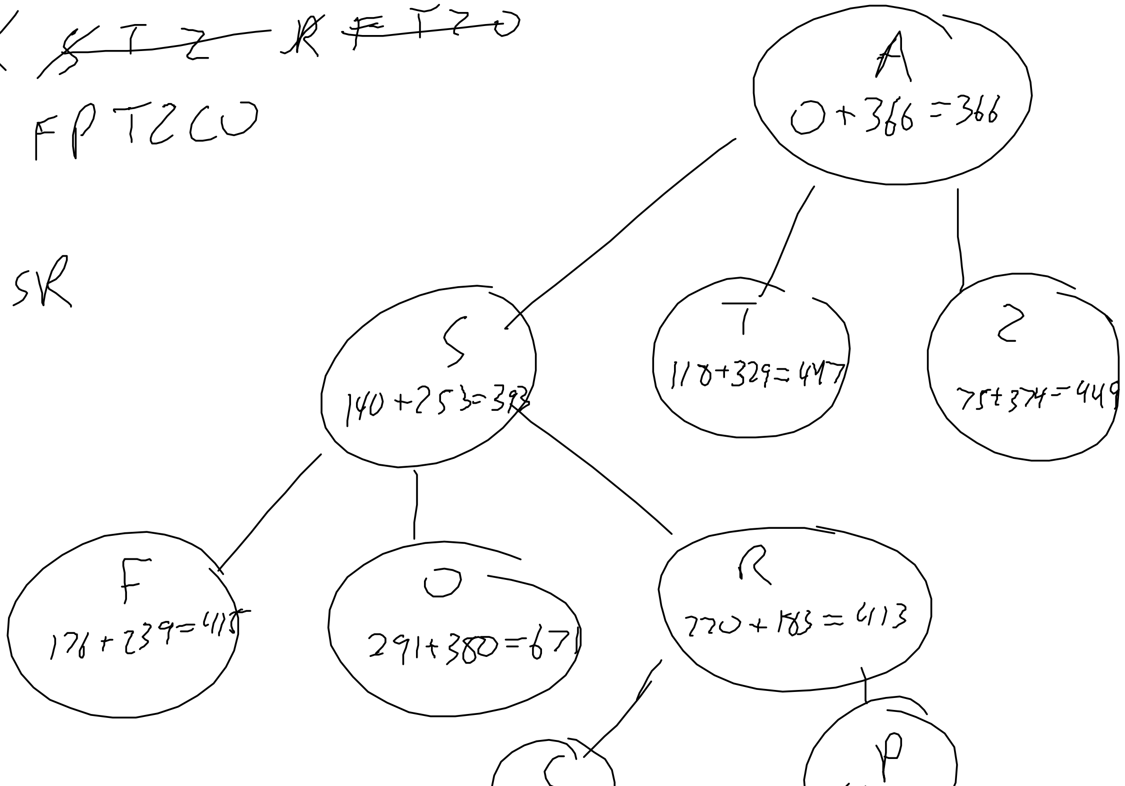
Estimate of n to nearest goal state

Be optimistic

open: ~~A~~ ~~S T Z~~ ~~R~~ ~~F T Z O~~

FPTZCO

closed: A SR


A
0+366=366
S
140+253=393
T
118+329=447
Z
75+374=449
F
176+239=415
O
291+380=671
R
220+193=413
C
526
P
417