Artificial Intelligence
Search: summary&exercises

Peter Antal
antal@mit.bme.hu

Reminder („symbols&search”): single state problem formulation

- A problem is defined by:
  - An initial state, e.g. *Arad*
  - Successor function *S(X)*= set of action-state pairs
    - e.g. *S(Arad)={<Arad $\rightarrow$ Zerind, Zerind>,…}*

  intial state + successor function = state space
  - Goal test, can be
    - Explicit, e.g. *x='at bucharest'*
    - Implicit, e.g. *checkmate(x)*
  - Path cost (additive)
    - e.g. sum of distances, number of actions executed, …
    - *c(x,a,y)* is the step cost, assumed to be >= 0

A solution is a sequence of actions from initial to goal state.
Optimal solution has the lowest path cost.

Reminder: tree-search

function TREE-SEARCH(*problem,fringe*) **return** a solution or failure
  fringe $\leftarrow$ INSERT(MAKE-NODE(INITIAL-STATE[*problem*]), *fringe*)
  loop do
    if EMPTY?(*fringe*) **then return** failure
    node $\leftarrow$ REMOVE-FIRST(*fringe*)
    if GOAL-TEST[*problem*] applied to STATE[*node*] succeeds
      then return SOLUTION(*node*)
    fringe $\leftarrow$ INSERT-ALL(EXPAND(*node*, *problem*), *fringe*)

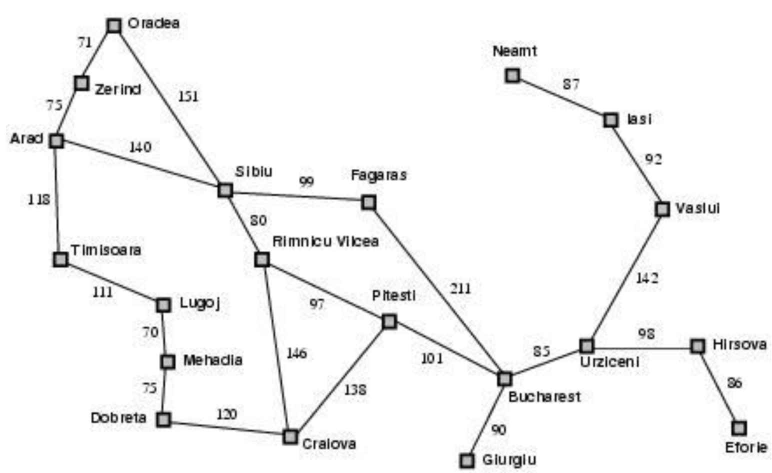
A strategy is defined by picking *the order of node expansion*

Reminder: main properties of uninformed search algorithms

Criterion	Breadth-First	Uniform-cost	Depth-First	Depth-limited	Iterative deepening	Bidirectional search
Complete?	YES*	YES*	NO	YES, if $l \geq d$	YES	YES*
Time	b^{d+1}	$b^{C^*/e}$	b^m	b^l	b^d	$b^{d/2}$
Space	b^{d+1}	$b^{C^*/e}$	bm	bl	bd	$b^{d/2}$
Optimal?	YES*	YES*	NO	NO	YES	YES

Route finding problems

Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Dobreta	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



- h_{SLD}=straight-line distance heuristic.
- Check animations!

Questions 1

What are the properties of bidirectional search? What is a necessary condition for its application?

Explain a search method with linear space complexity!

What is an admissible heuristics? Give an example and a counterexample.

Explain the properties of the iterative deeping search.

Define the components of a single-state problem in the „informative” case

Explain methods to construct heuristic functions and to improve them by combination.

Define the elements of the single-state problem, both in the uninformed and informed cases and explain a corresponding data-structures (10 p)

Q 2

- What is the best search method, if we have no heuristic function, neither a priori knowledge about the depth of the solution, and the state space is infinite? Why (what are the properties)?
- What are the problems with local search (optimization) methods? Is there any global optimization method?
- Give examples for search methods with different space and time complexity (use the maximum branching factor b, depth of least cost solution d, and maximum depth of diameter m).

Q 4 Equality, optimality

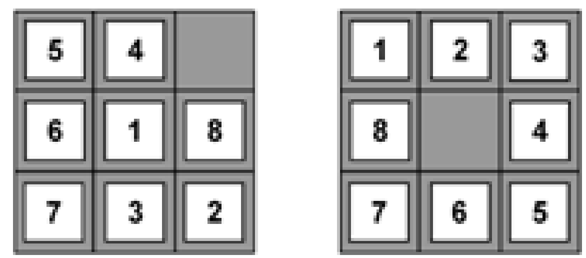
- Decide the validity of the following:
  - (a) Solution (A*) = Solution (uniform cost search)?
  - (b) Solution (A*) = Solution (greedy)?
  - (c) Depth of solution(iterative deepening depth first) = Depth of solution (breadth-first-search)?
  - (d) Solution (depth-first-search) = Solution (greedy)?
- And for the costs?

Heuristics

- If h1, h2, h3 heuristics are admissible separately, then prove that the following combinations are admissible or not (6 points):
  - (a) $h = h1+h2+h3$,
  - (b) $h = (h1+h2+h3)/2$,
  - (c) $h = (h1+h2+h3)/3$.

Heuristics 2

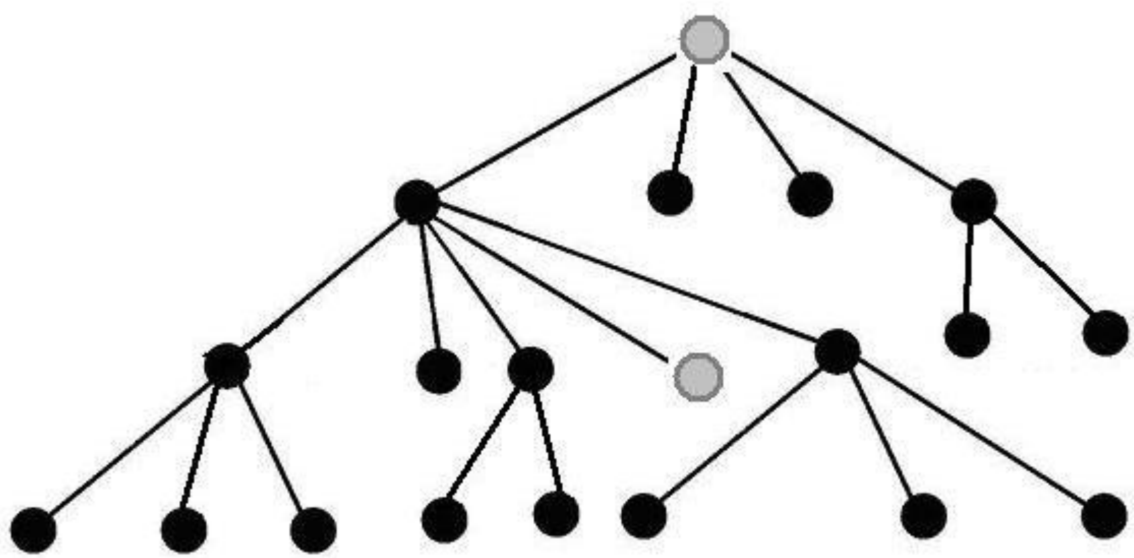
- Three heuristics was proposed for the puzzle game based ont he following functions:
  - $h_1(\text{tile}) = |\Delta x|+|\Delta y|$,
  - $h_2(\text{tile}) = \max(|\Delta x|, |\Delta y|)$, and
  - $h_3(\text{tile}) = \text{sqrt}\ ((\Delta x)^2+(\Delta y)^2)$.
- The „estimed distance” of a configuration from a goal is given by the sum Σ_{tiles} h(tile). Compute the „estimed distance” from the start state on the left to the goal state on the right, using all three heuristic function.



- Characterize the heuristics with the „h_x admissible”, „h_x not admissible”, „h_x dominates h_y”.

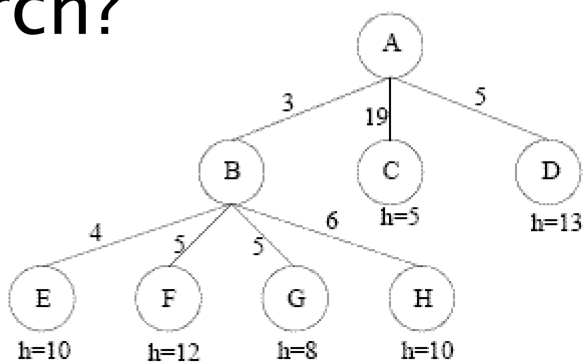
Effective branching factor

- A search tree is shown below, gray nodes indicate the start and end nodes, black nodes were inserted into the queue. What is the branching factor, effective branching factor, and what is the approximate value of the effective branching factor in the following case?



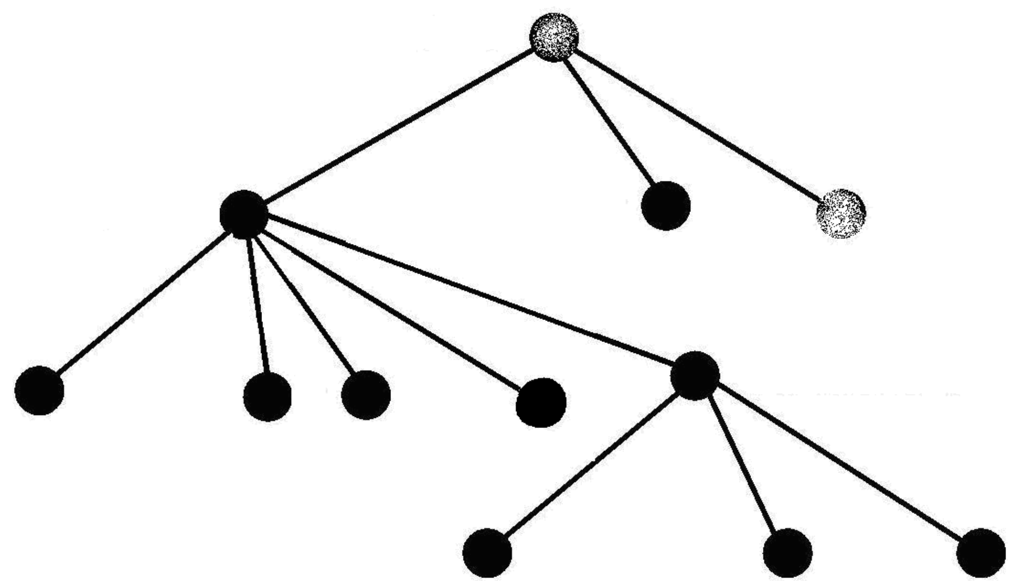
Search

- In the search tree below the costs of the movements and the heuristic estimates are shown. Which node will be expanded next in
- (i) in breadth-first search?
- (ii) in uniform cost search?
- (iii) in greedy search?
- (iv) in A* search?



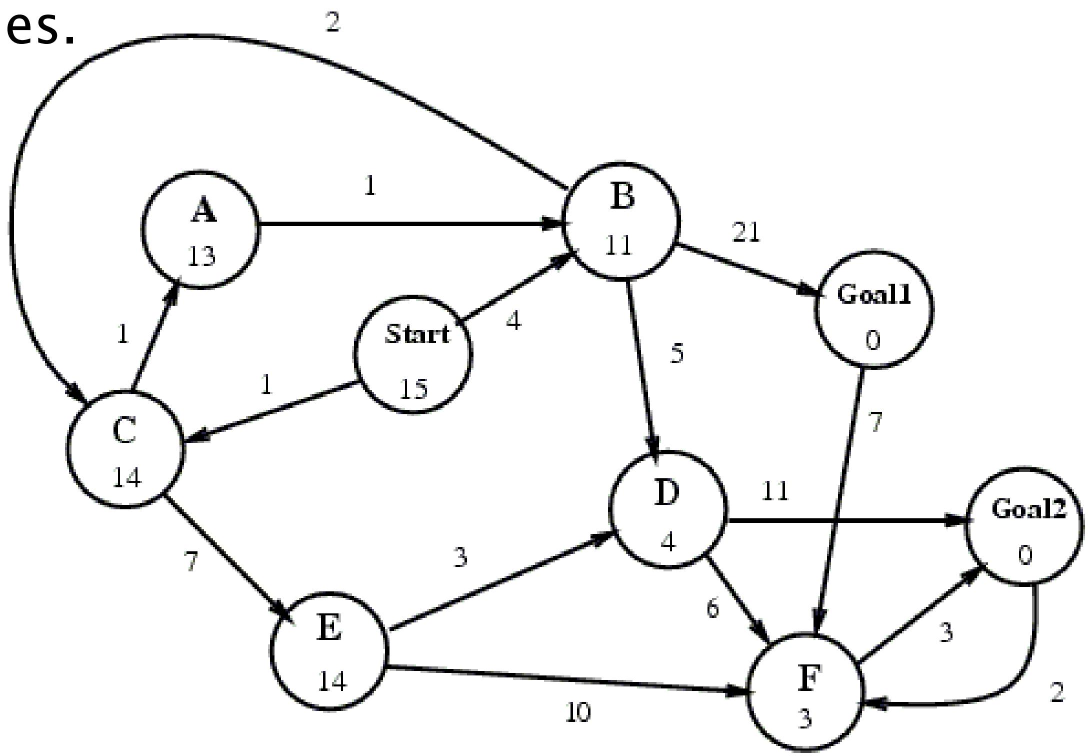
DF vs DL search

- Apply the depth-first search and the depth-limited search on the tree below and number the nodes with the order of their goal-test and expansion(s). The initial state is the root and the only goal state is the other shaded node. Assume that the expanded nodes are inserted into the queue form left to right.



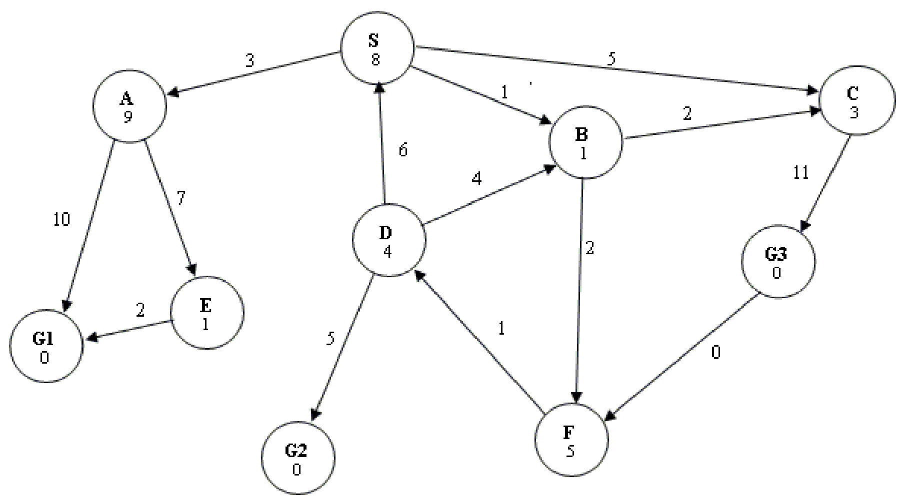
Informed search

- Which goal state will be the result of (a) greedy, and (b) uniform cost search methods? The heuristic function values are given in the nodes, the costs of the movements are indicated on the edges.



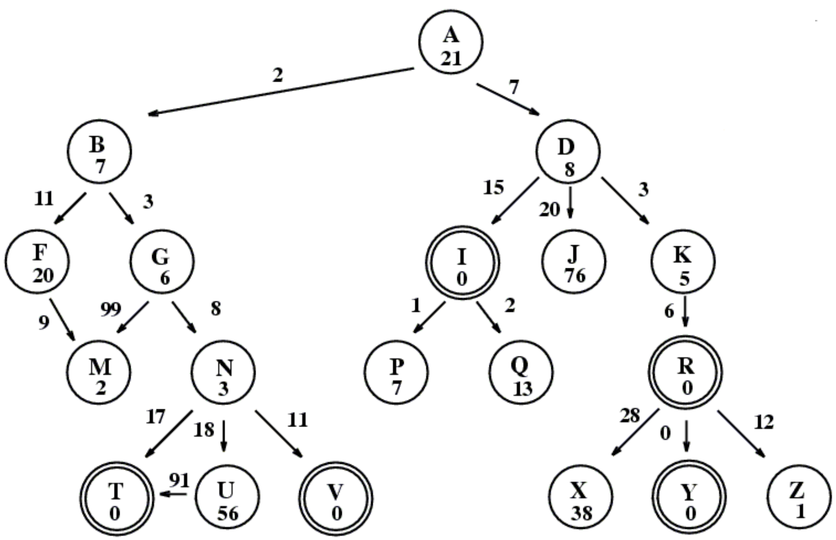
Informed search 2

- Find an optimal solution using A* in the search tree below. S denotes the initial state, G indicates goal states, the heuristic values of the states are shown in the nodes, costs of actions are indicated along the edges.



Informed search 3

- Apply (a) the depth-first search, (b) the uniform-cost search, and (c) the greedy search from state A and determine the resulting goal state (goal states are denoted with double circles). For each search method describe its main properties (time complexity, space complexity, completeness, optimality).



Informed search 4

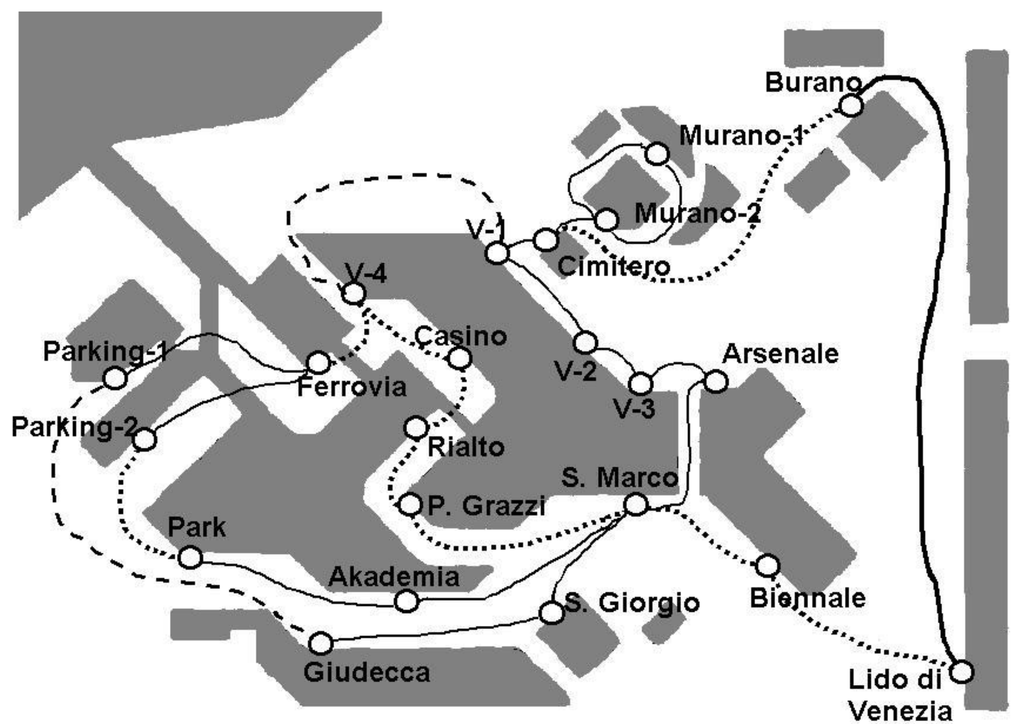
- Apply the algorithm **A*** from A to B and fill the cells according to the scheme on the right (**h** the value of heuristic, **g** denotes the minimal cost of the path, **f** is th total cost, and **a** is the altitude of the cell). The applied heuristics is the Manhattan distance (minimum number of steps using only the four neighbours). The real cost to a neighbouring cell is 1 + Δ, where Δ is the distance between the altitudes if we step to higher place, otherwise Δ is 0. Indicate the optimal path on the map!

0	0	0	2	0
A 0	1	1	2	0
0	0	0	2	0
0	0	0	2	0
1	1	1	2	B 0
0	0	0	2	0

h	g
alt.	f

Search 5

- You have to travel from the **Lido** to the station (**Ferrovia**) in Venice. The only available vehicle of transport is the *vaporetto* (water-bus), whose simplified network is shown in the figure. The (time) cost is 4 for a **bold black** section, 3 for a **dotted** section, 2 for a **dashed** section and 1 (fastest) is for the sections indicated with normal black line. Select an appropriate search strategy and find the fastest route. Number the stations as they are expanded/checked by the algorithm. (8 p)



Summary

- Idea/pseudocode of the search methods
- Properties of applicability
- Demonstrative application in toy problems