Imperial College London

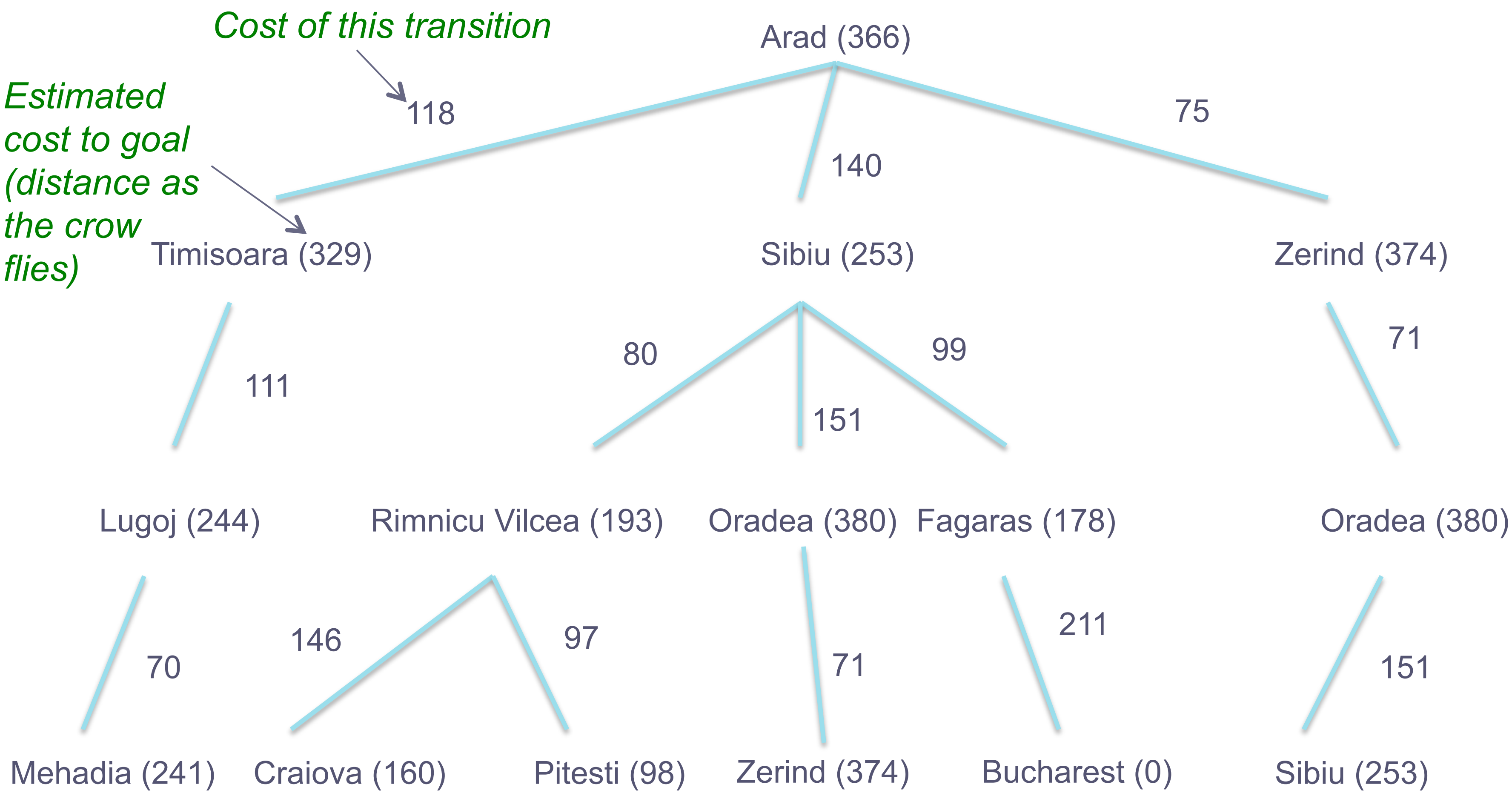
Informed Search

Murray Shanahan

Overview

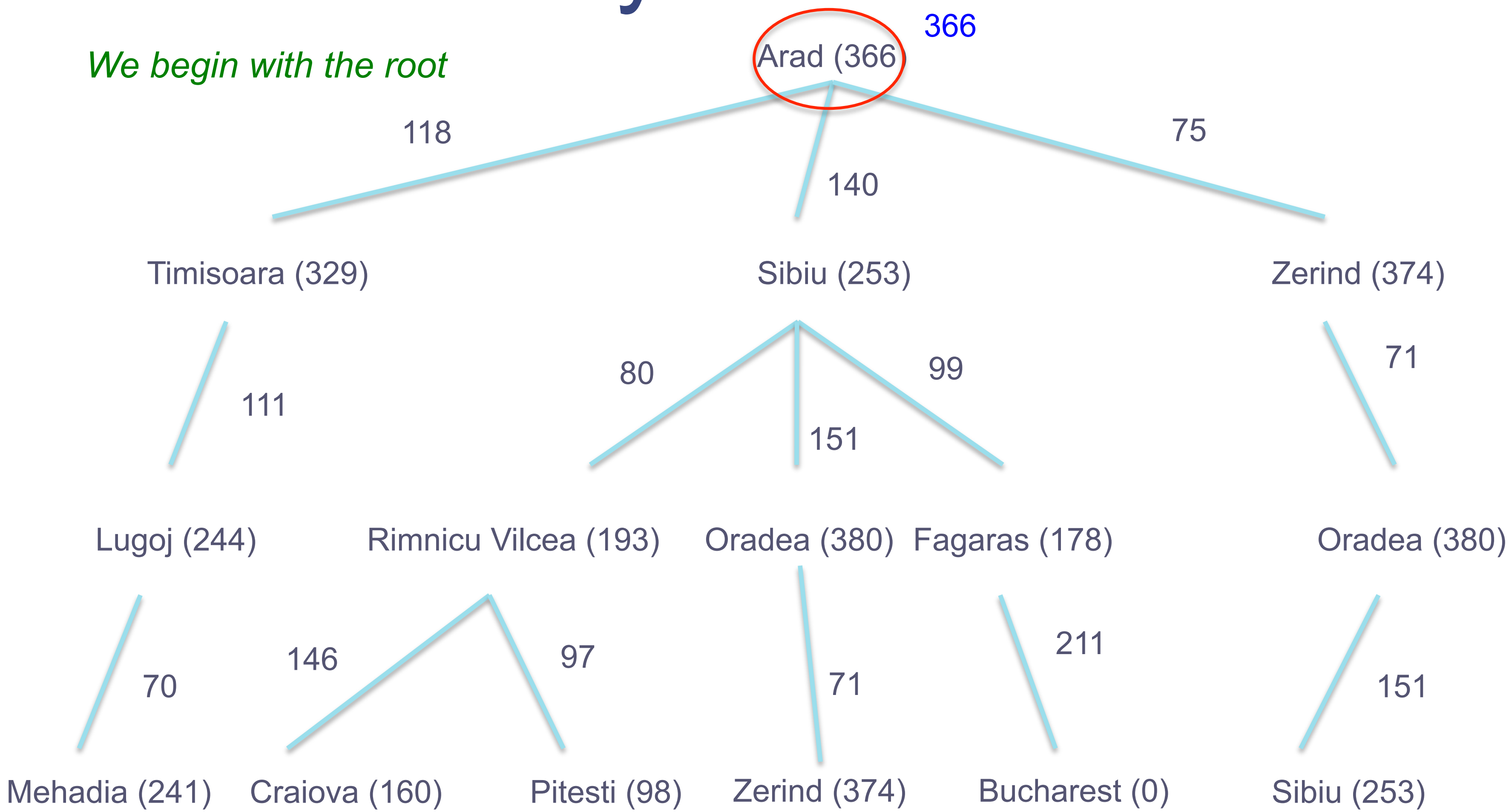
- Greedy best-first search
- The A* algorithm

Tree with Costs and Estimates

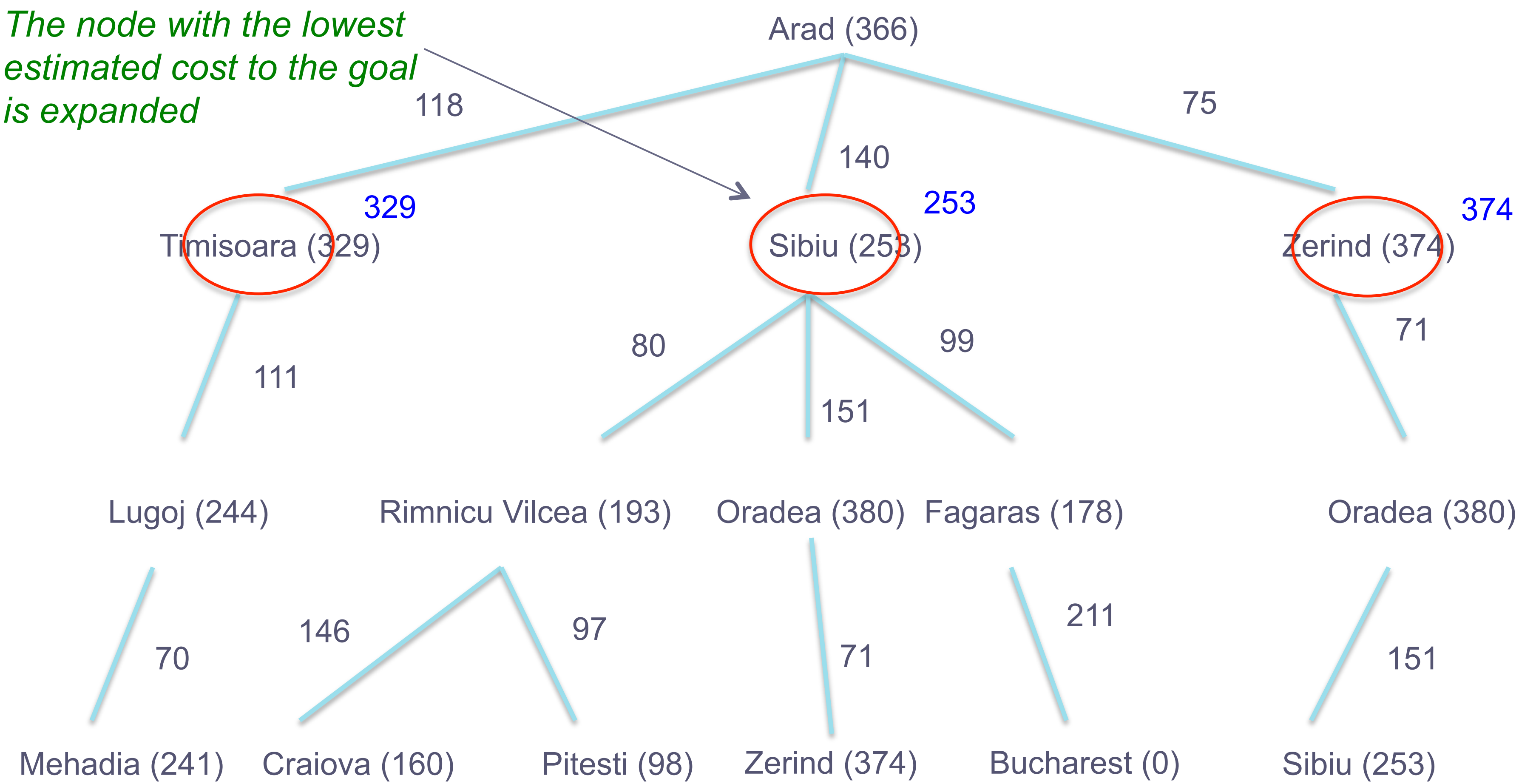


Greedy Best-first Search

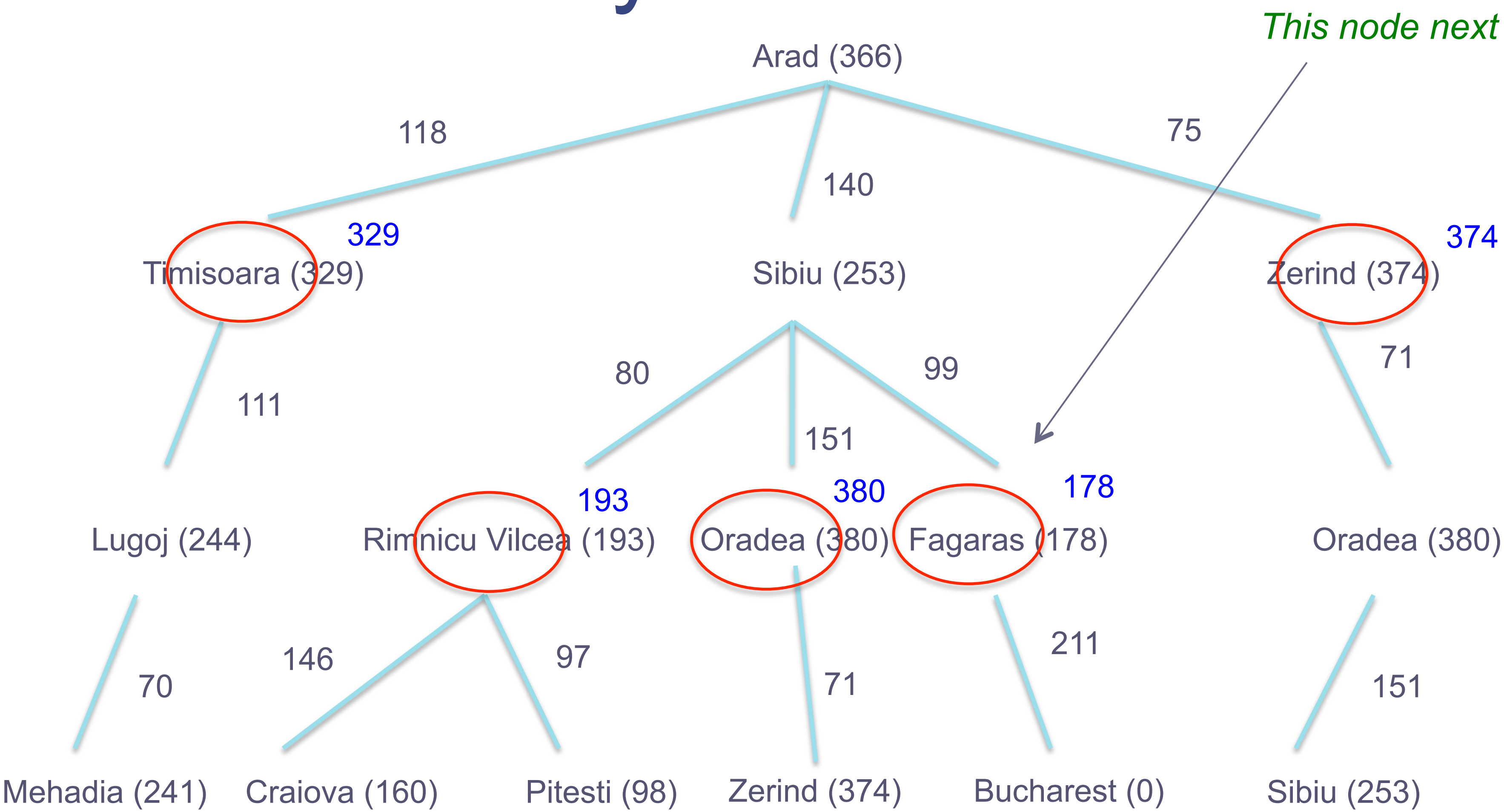
Greedy Best-first 1



Greedy Best-first 2

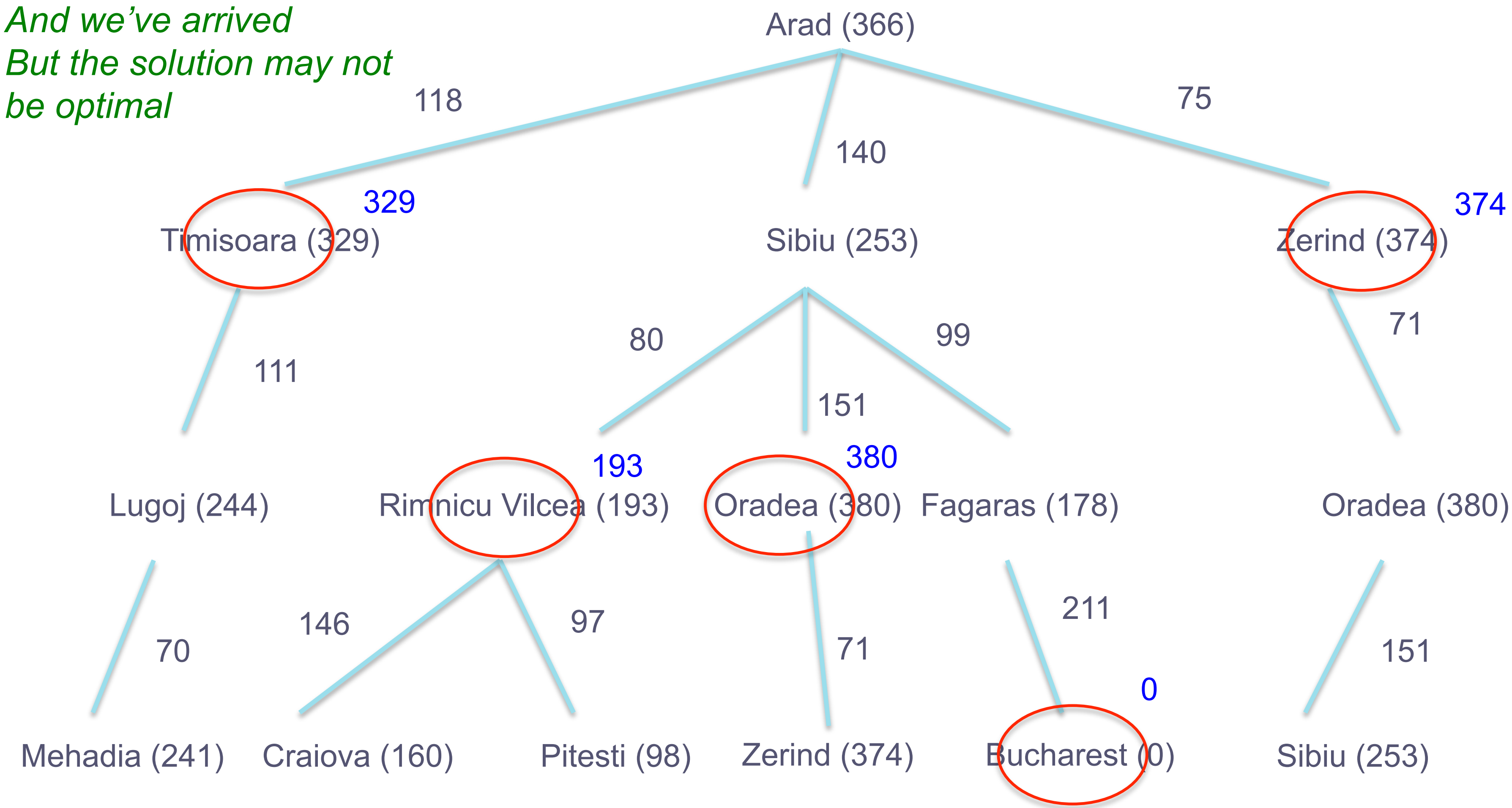


Greedy Best-first 3



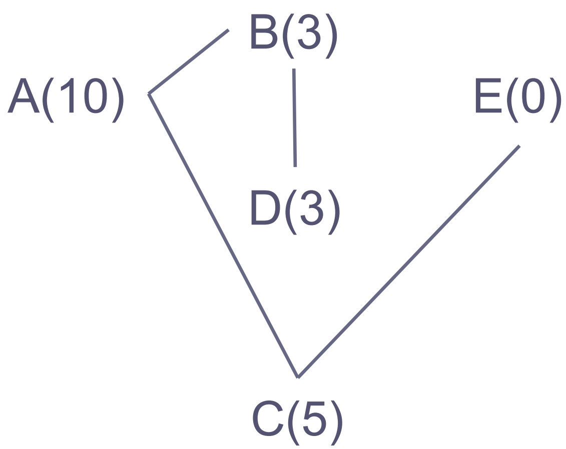
Greedy Best-first 4

And we've arrived
But the solution may not be optimal



Properties of Greedy Best-first

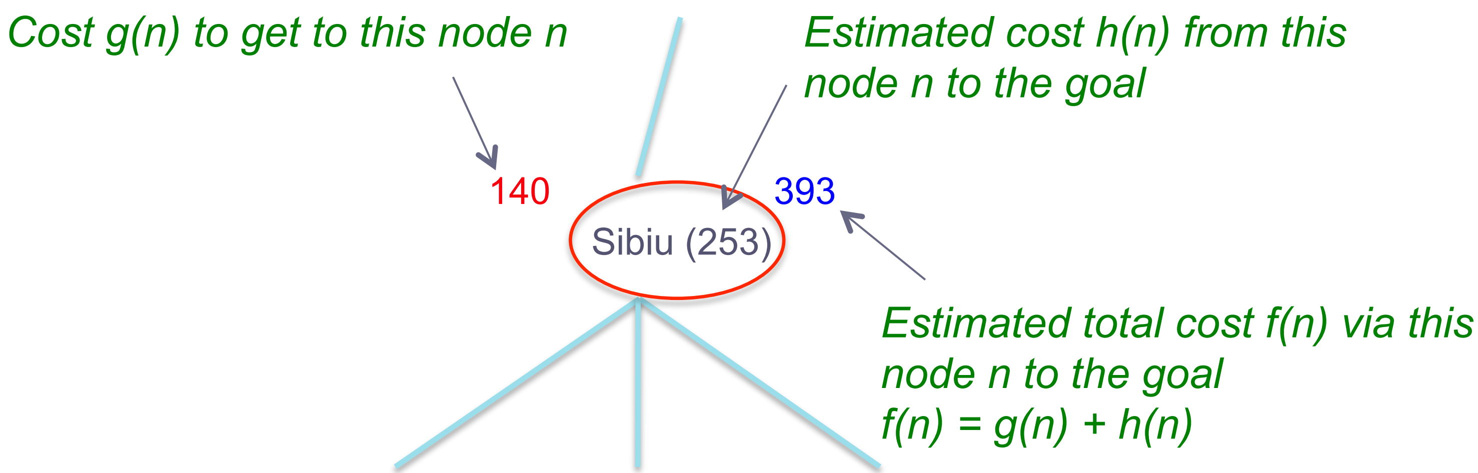
- Not guaranteed to find optimal solution
- Not guaranteed to find a solution if one exists, because it can get stuck in a loop
- Suppose A is the initial state and E is the goal
- Greedy best-first will go to node B, then oscillate forever between nodes B and D
- It will never try C



The A* Algorithm

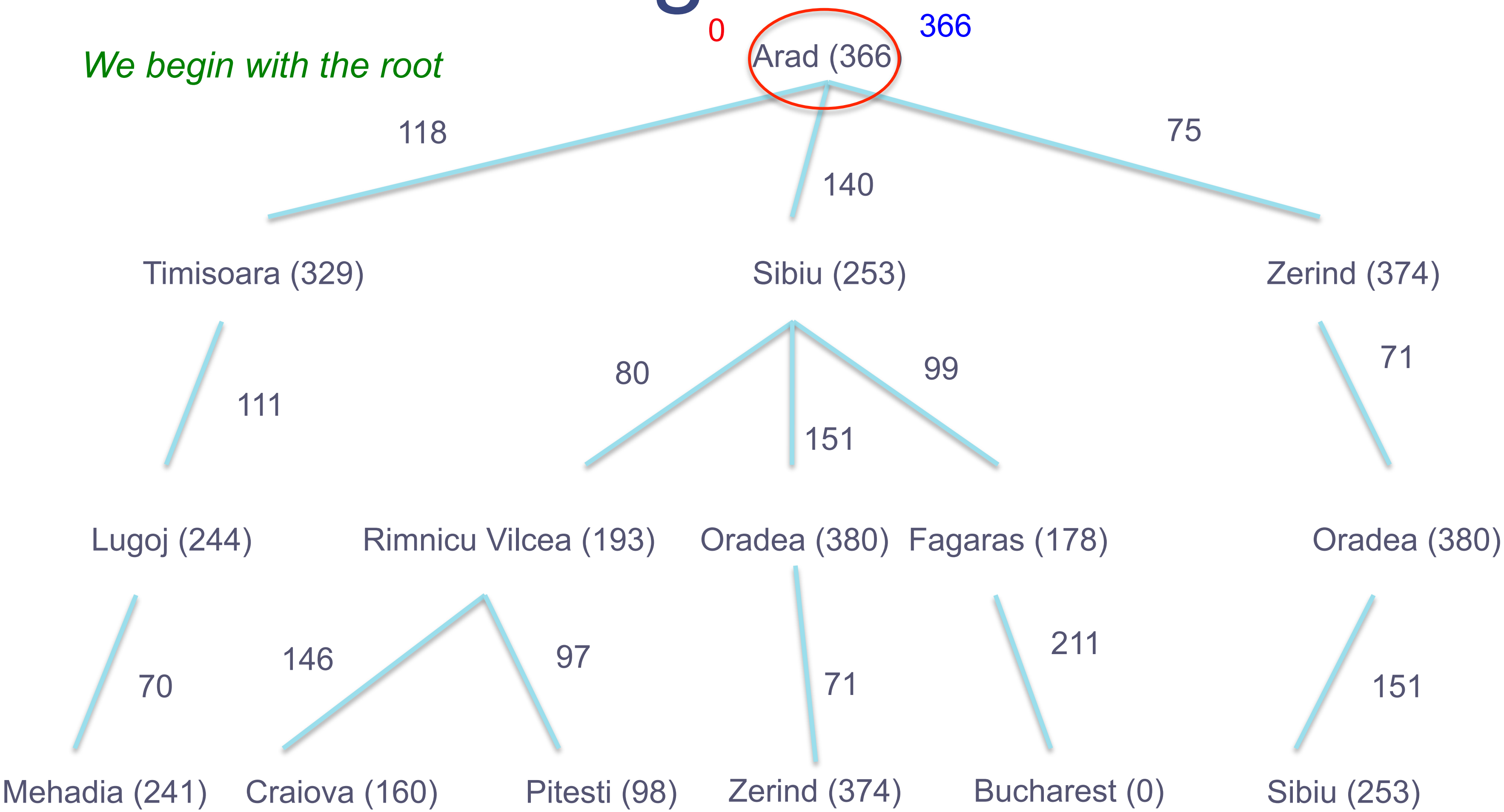
The A* Heuristic

- The A* algorithm uses a heuristic (an *evaluation function*) that combines the cost of the path to a node with the estimated cost from that node to the goal

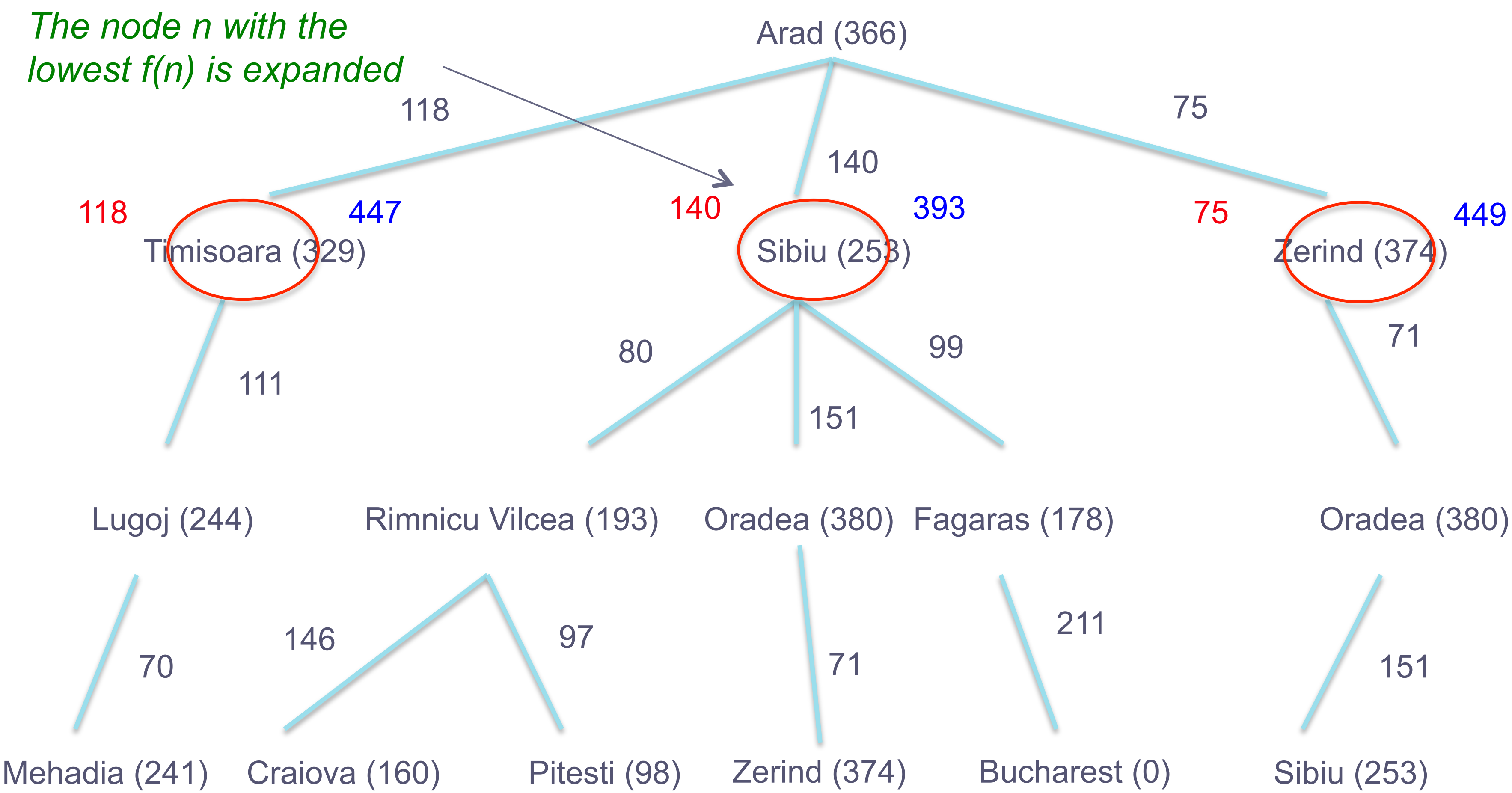


A* Algorithm 1

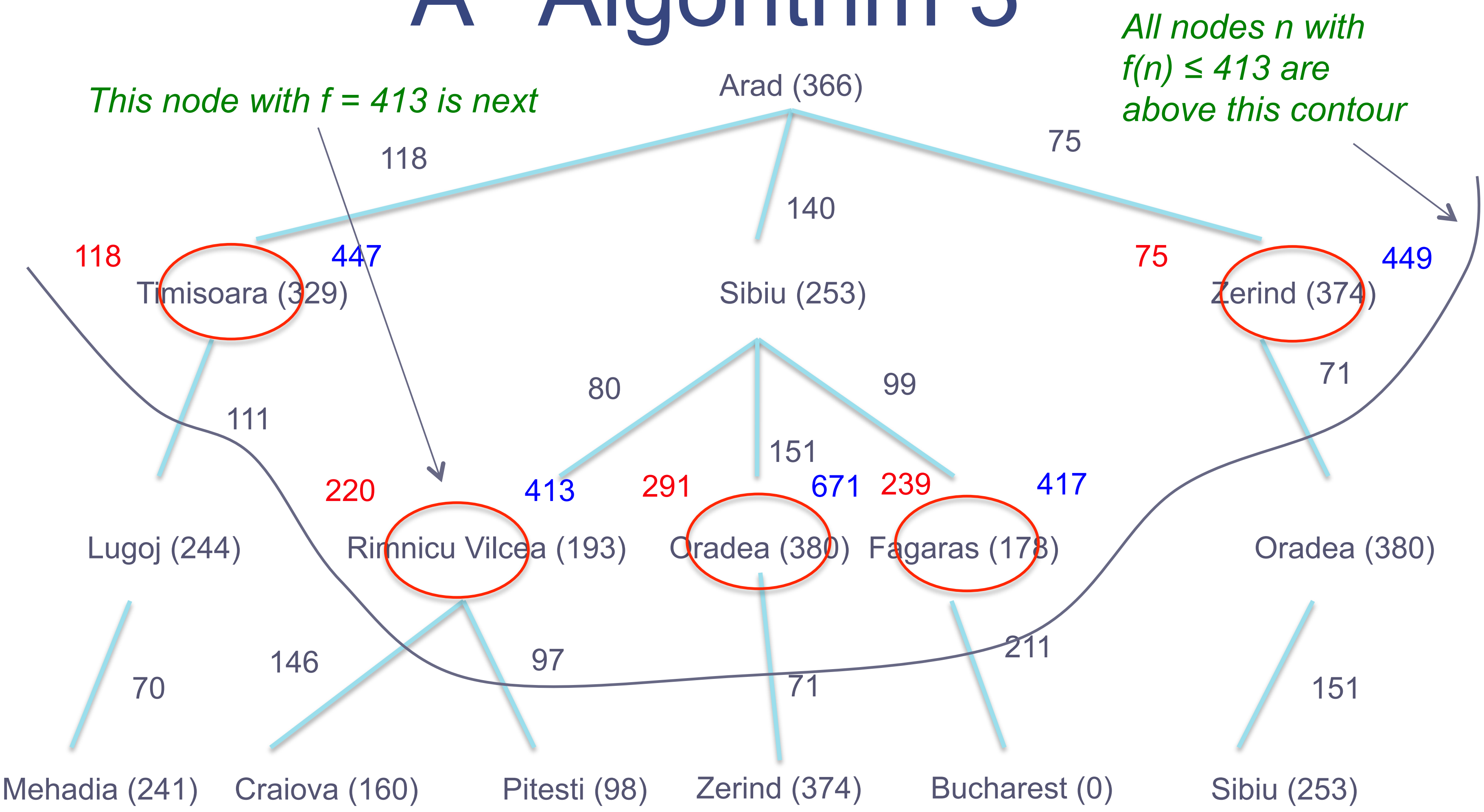
We begin with the root



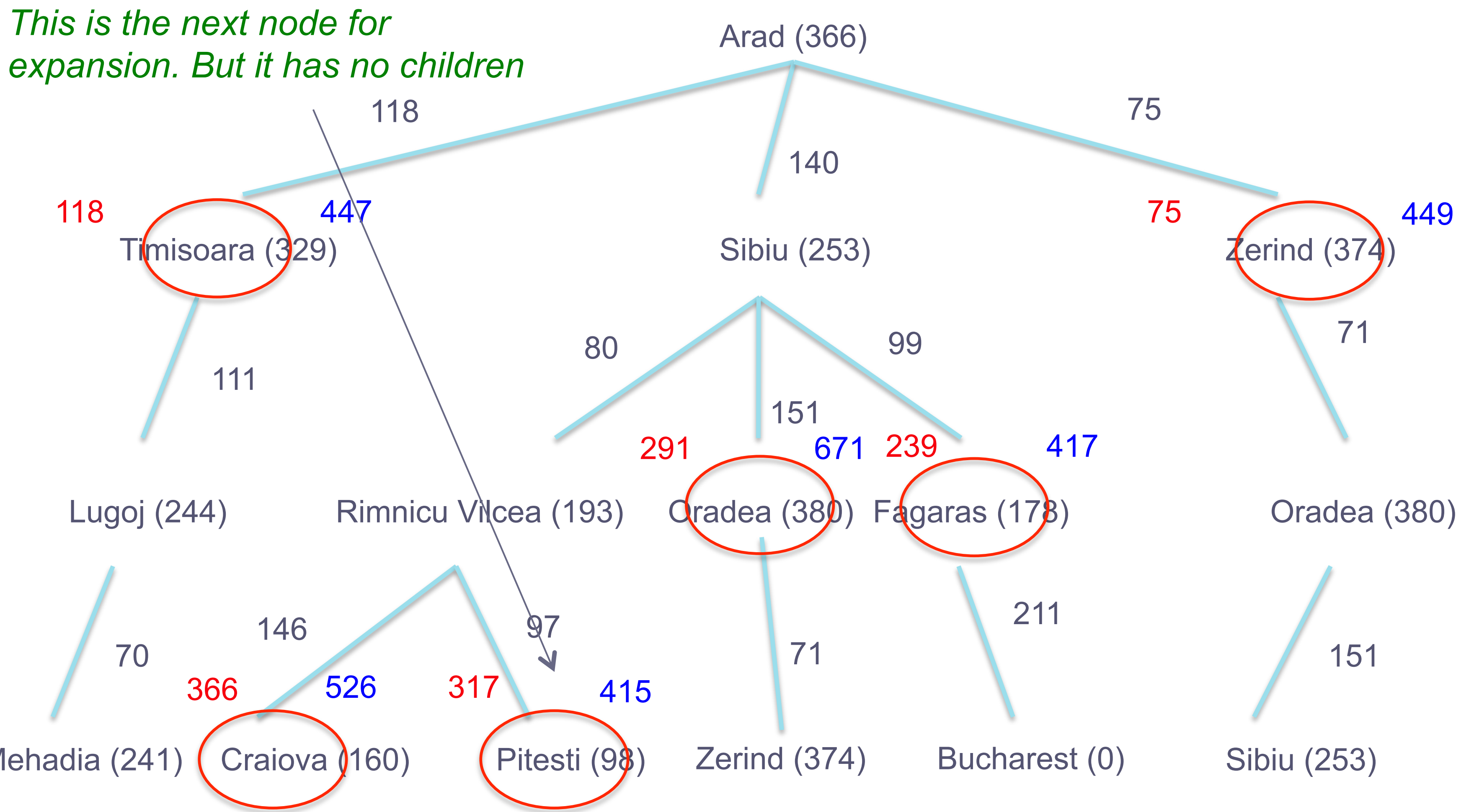
A* Algorithm 2



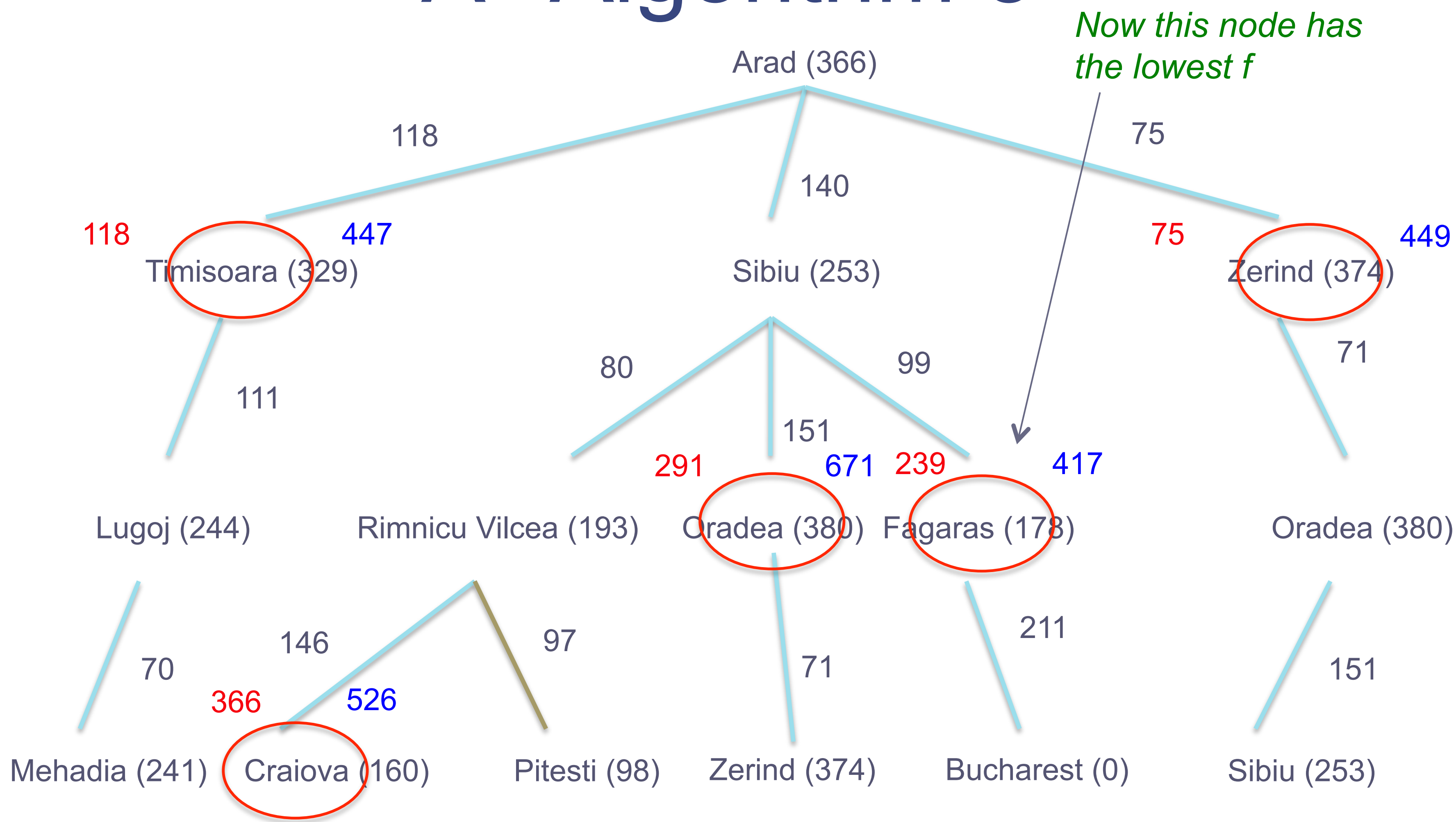
A* Algorithm 3



A* Algorithm 4

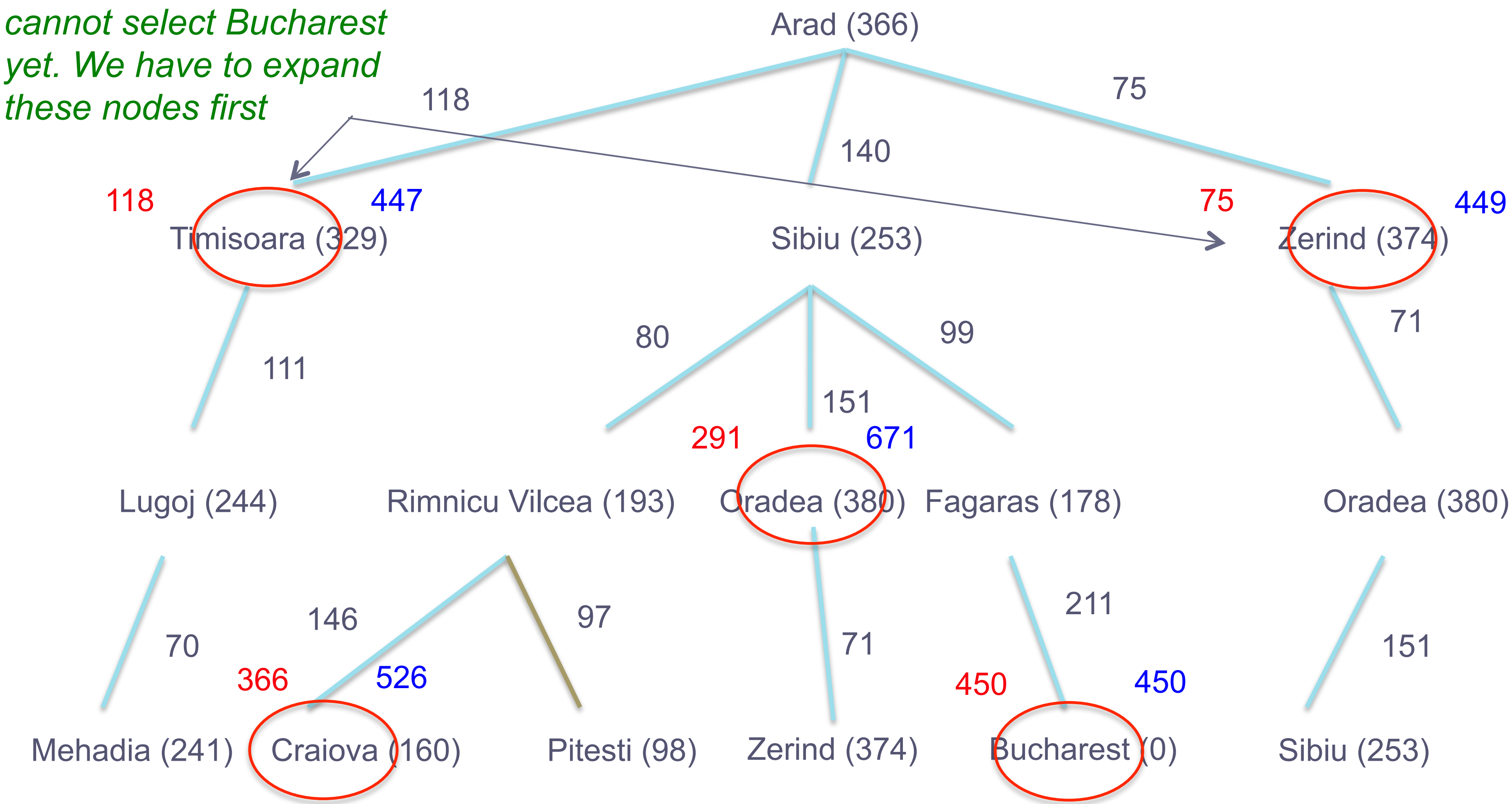


A* Algorithm 5



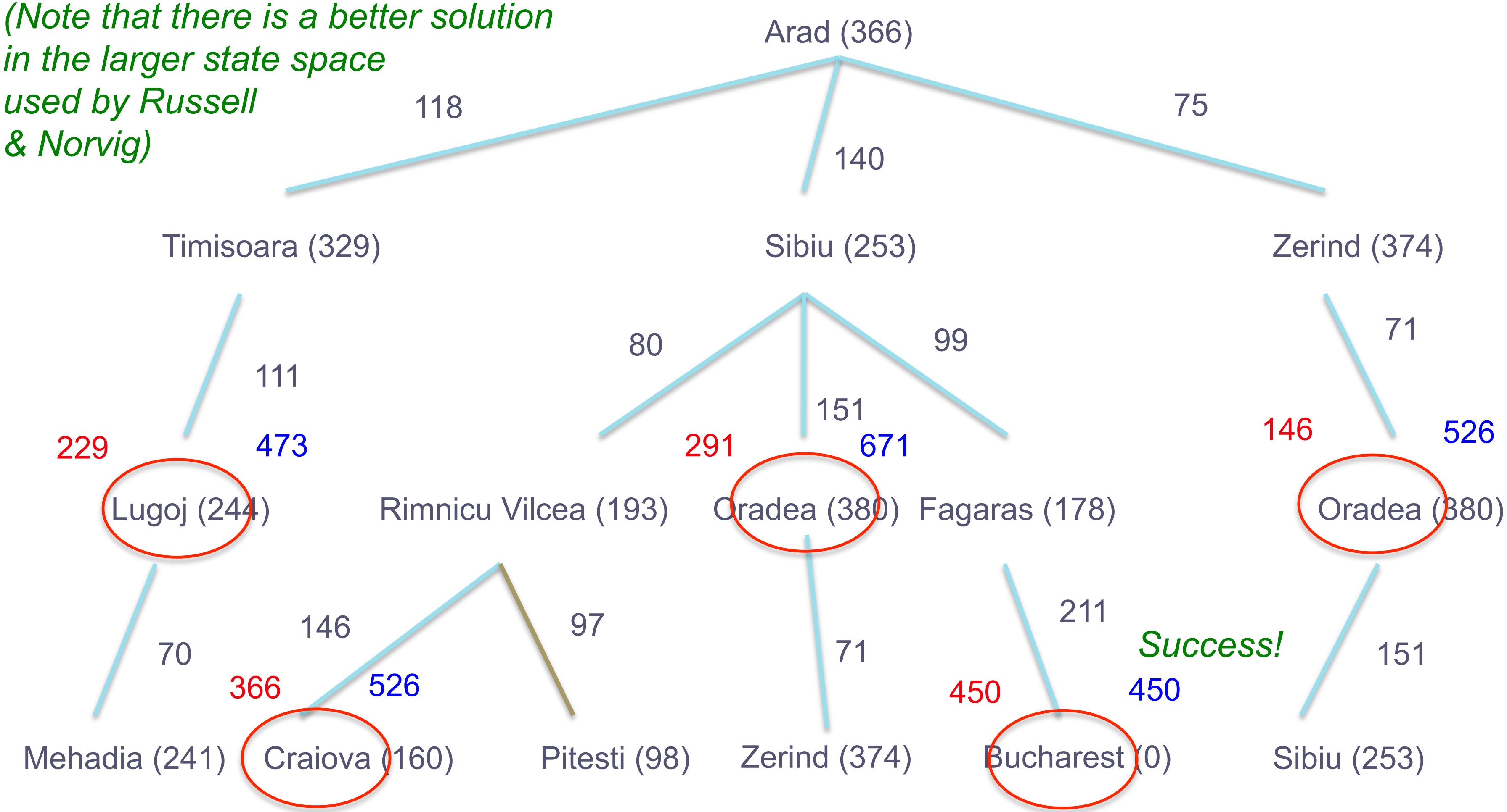
A* Algorithm 6

Success? Not quite. We cannot select Bucharest yet. We have to expand these nodes first



A* Algorithm 7

(Note that there is a better solution in the larger state space used by Russell & Norvig)



Properties of A*

- Guaranteed to find a solution if one exists, as long as there are only finitely many nodes n with $f(n) \leq f(goal)$
- Guaranteed to find the optimal solution
- But must use an *admissible heuristic*
- $h(n)$ is admissible iff for all nodes n, $h(n) \leq h^*(n)$, where $h^*(n)$ is the true cost to the goal from n
- Example: distance as crow flies is an admissible heuristic, because it is always less-than-or-equal to the true distance travelled

Optimality of A*

- Goal node G found by A* is guaranteed to have the lowest path cost C
- Proof (by contradiction):
  - Suppose A* selects some goal node G2 such that $g(G2) > C$
  - Let n be an unexpanded node on a lowest cost path to a goal node
  - h is admissible, so $f(n) = g(n)+h(n) \leq C$
  - But the path to G2 is sub-optimal, so $f(G2) = g(G2)+0 > C$
  - So $f(n) < f(G2)$
  - Therefore A* must select n before G2

The Proof in a Picture