

Artificial Intelligence 2007 Spring

Homework 2

Due date: March 29, 2007

April 10, 2007

1. **AIMA[4.1]**

Trace the operation of the A* search algorithm applied to the problem of getting to Bucharest from Lugoj using the straight-line distance heuristic. That is, show the sequence of nodes that the algorithm will consider and the f , g , and h score for each node.

Ans.

The steps using TREE-SEARCH are shown below as Figure 1 to Figure 11. You can also use GRAPH-SEARCH in this problem if you show that the heuristic function here is consistent.

2. **AIMA[4.8]**

The traveling salesperson problem (TSP) can be solved via the minimum spanning tree (MST) heuristic, which is used to estimate the cost of completing a tour, given that a partial tour has already been constructed. The MST cost of a set of cities is the smallest sum of the link costs of any tree that connects all the cities.

- a. Show that this heuristic can be derived from a relaxed version of the TSP.
- b. Show that MST heuristic dominates straight-line distance.
- c. Write a problem generator for instances of the TSP where cities are represented by random points in the unit square.
- d. Find an efficient algorithm in the literature for constructing the MST, and use it with an admissible search algorithm to solve instances of the TSP.

Ans.

- (a) If we relax the constraint for TSP such that each city can be visited more than one time (that is, there may be some cities visited more than one time) and the cost for repeated edges are not count, we can get a solution for MST problem.
- (b) MST dominates straight-line distance because the shortest distance from any two cities u , v is their straight-line distance. Moreover, let $G_{MST}(V, \bar{E})$ be the solution of MST problem given

the instance G . The cost from u to v for MST will be equal to the straight-line distance only if $(u, v) \in \bar{E}$. Thus MST heuristic dominates straight-line distance.

(c) A simple generator can be:

```

Generator(Integer Nums){
    POINTS = nil
    N = 0
    while(N < Nums){
        (X,Y) = (Random(0,1), Random(0,1))
        if ((X,Y) ∉ POINTS){
            POINTS ← (X,Y)
            N = N + 1
        }
    }
    return POINTS
}

```

(d) Any MST algorithm is acceptable. You can use the MST algorithm as an oracle. The detailed description of MST algorithm can be omitted in this problem. Figure 12 is an instance for TSP, and Figure 13 to Figure 18 are the steps using A* algorithm with MST as its heuristic function.

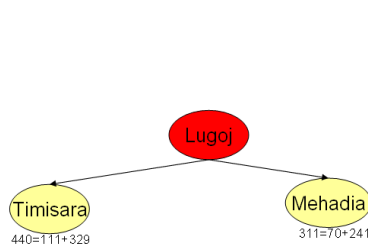


Figure 1: Start from Lugoj

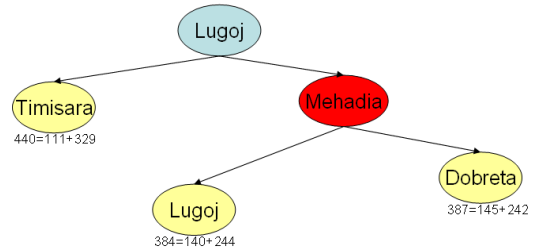


Figure 2: Choose Mehadia

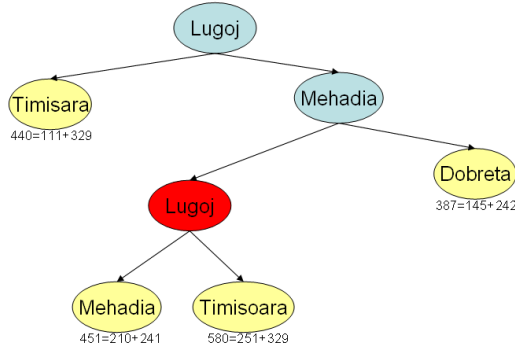


Figure 3: Lugoj here is obvious a wrong choice

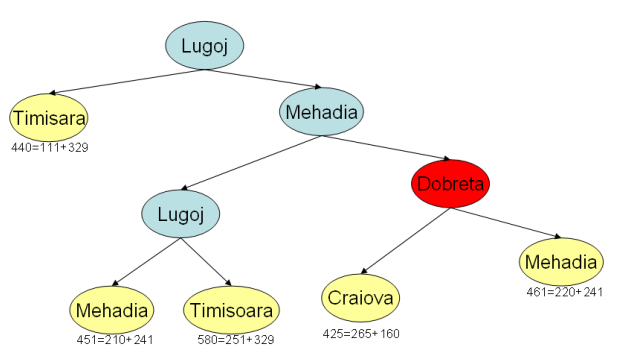


Figure 4: Return to Dobreta

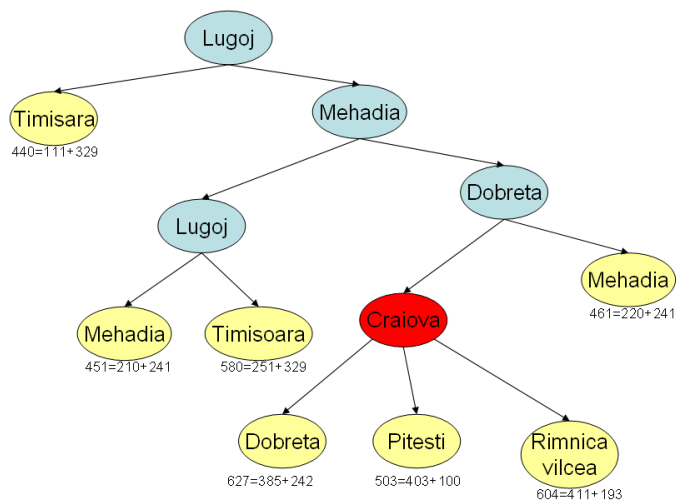


Figure 5: Go to Craiova

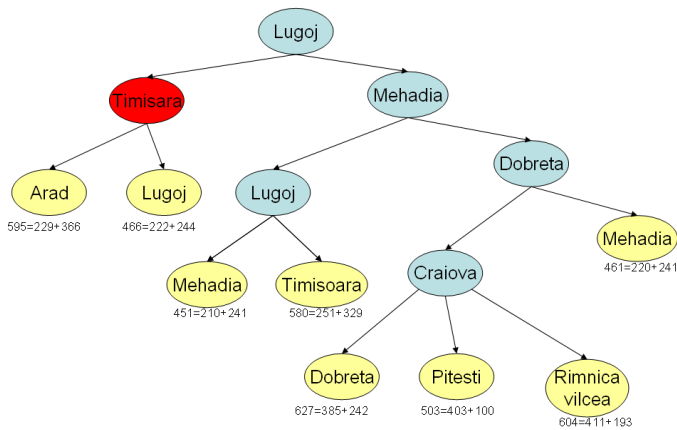


Figure 6: Go back to Timisoara

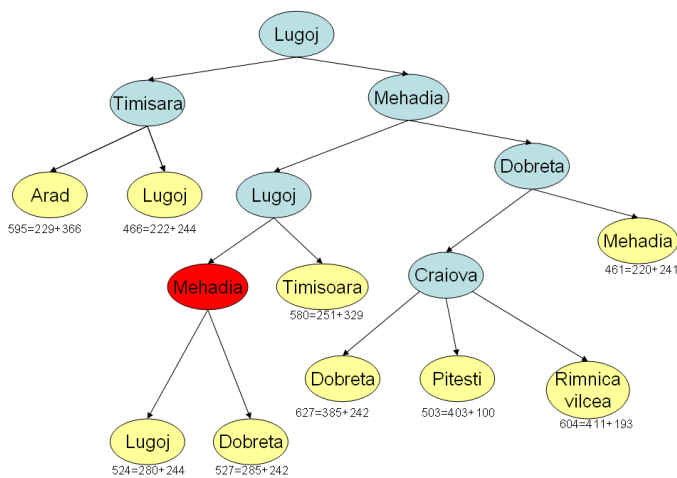


Figure 7: Mehadia revisited

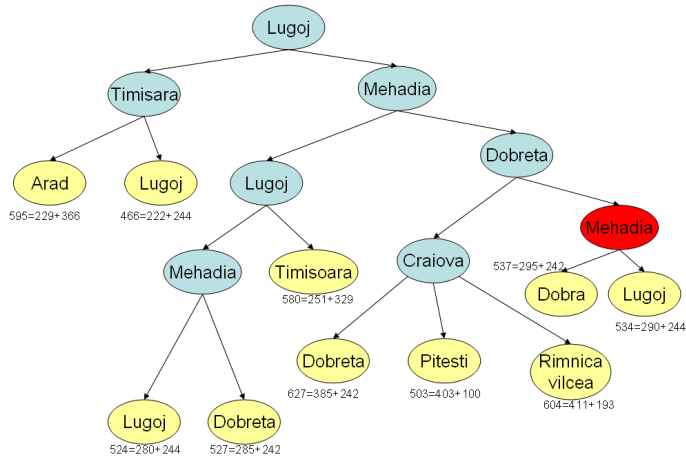


Figure 8: Another path to revisit Mehadia

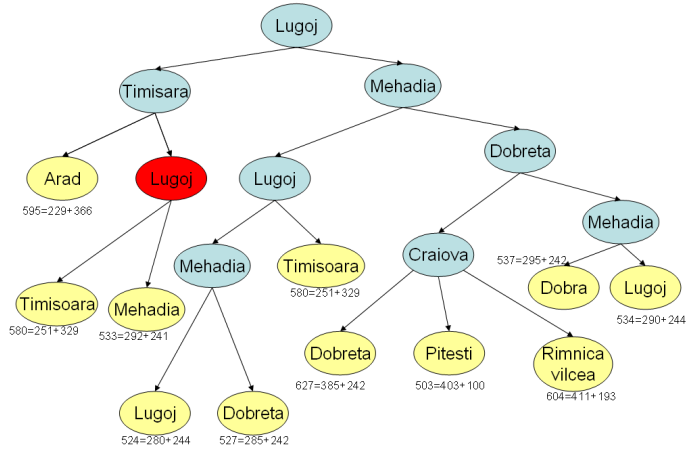


Figure 9: Go back to Lugoi again

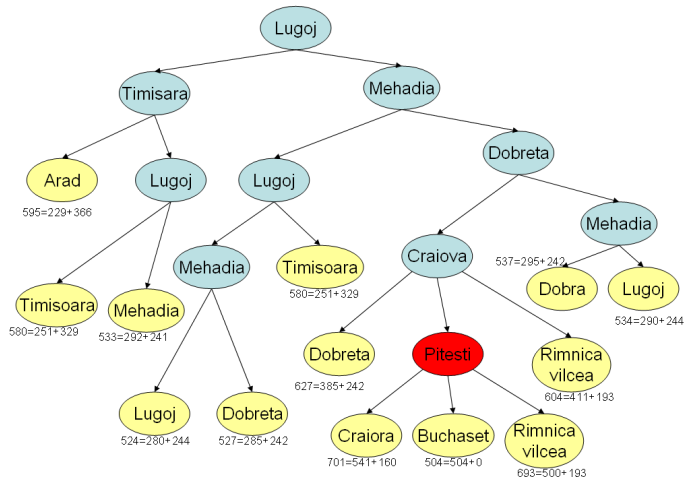


Figure 10: Finally reach Pitesti

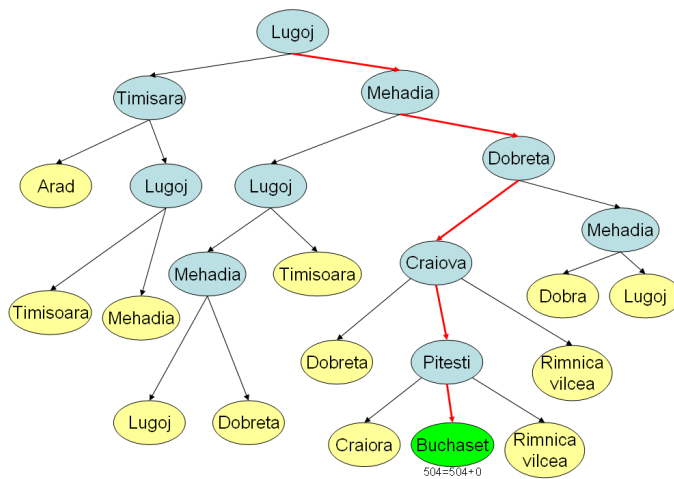


Figure 11: Arrive at the goal

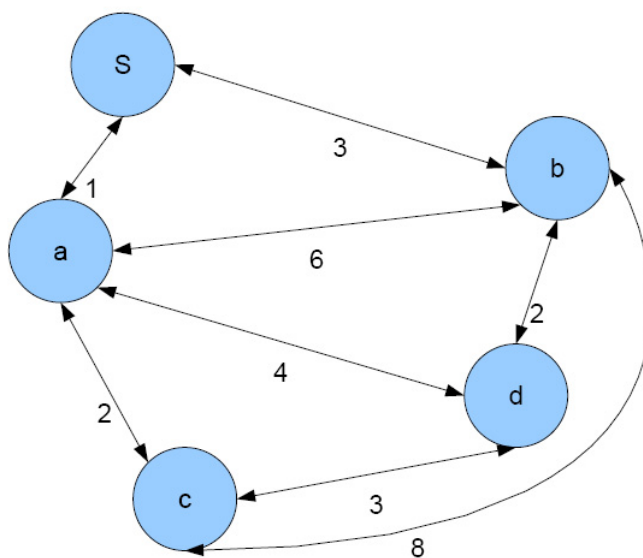


Figure 12: A TSP instance

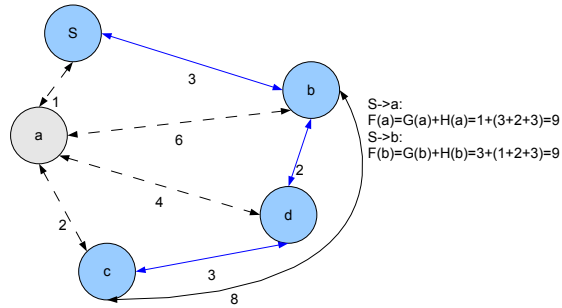


Figure 13: MST edges, after (s, a) is added into TSP path

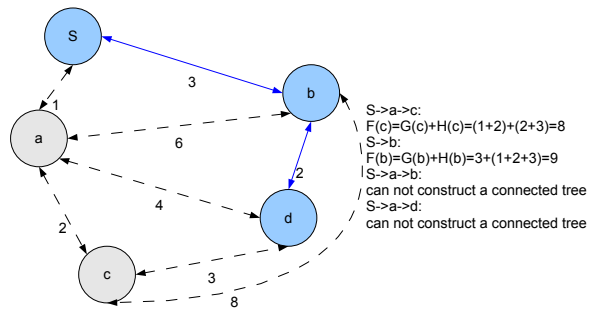


Figure 14: MST edges, after (s, a) , (a, c) is added into TSP path

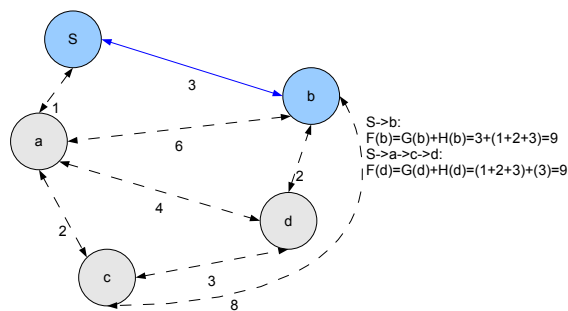


Figure 15: MST edges, after (s, a) , (a, c) , (c, d) is added into TSP path

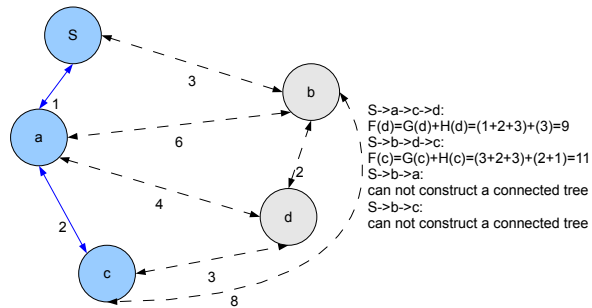


Figure 16: Try another path

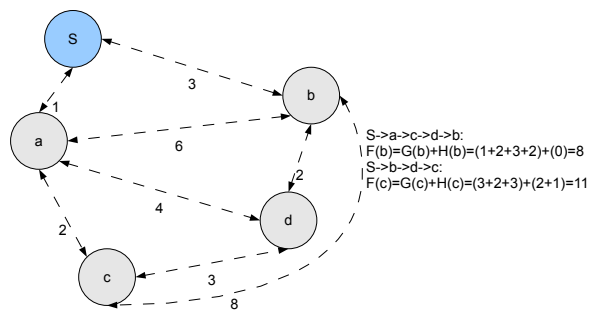


Figure 17: Return to previous choice

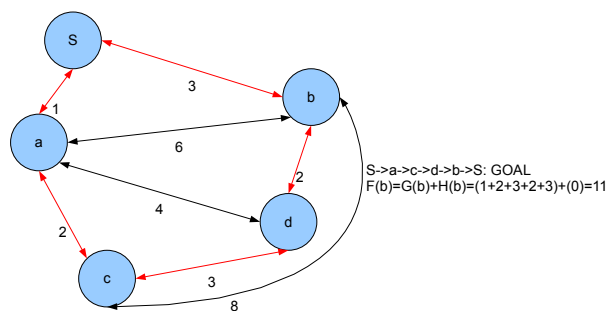


Figure 18: A TSP solution is found