A1Q1

May 20, 2022 10:31 PM

1 a) Suppose we have an undirected graph V.E where V is the set of all cities (verticies) and E is the set of all edges.

initial state $(S_c, \langle S_c \rangle)$

where S_c is the start city ($S_c \in V$) and $\langle S_c \rangle$ is the travel path.

goal state $(S_c, \langle S_c, \cdots, S_c \rangle)$, $S_c \in V$

with a condition that $\langle S_c, \cdots, S_c \rangle$ is a valid path in the graph. and it contains every node in V.

neighbour rules: Suppose the last node we visited is M, N is a neighbour if $MN \in E$ and N is not in the state. i.e. MN is an edge and we haven't visited node N before when we hasn't visited all nodes. If we have visited all nodes, then the neighbour is the start node.

cost: the cost is defined by the cost of the edge. Suppose the last node we visited is M and N is neighbour, then the cost to add N to the state is the weight of edge MN.

example:

initial state: $(W, \langle W \rangle)$

i.e: the salesperson is in the start city and hasn't visited any other cities.

goal state: $(W, \langle W, Path, W \rangle)$ where Path contains G,T,H,B

One example: $(W, \langle W, G, T, H, B, W \rangle)$

i.e. the salesperson has visited all other cities and return to the start city

neighbours: any city that the sales person has not visited before is a neighbour (because this is a complete graph). when the salesperson hasn't visited all cities. Otherwise the neighbour is W.

e.g: for state $(W, \langle W, G, T \rangle)$, the neighbours are B, H.

for state $(W, \langle W, G, T, H, B \rangle)$, the neigbour is W.

costs: the cost is defined by the distance between two cities.

b) cost(p) is the total cost of path p. i.e. the sum of the distances between adjacent cities on the path.

example: Suppose p = WGTB, then

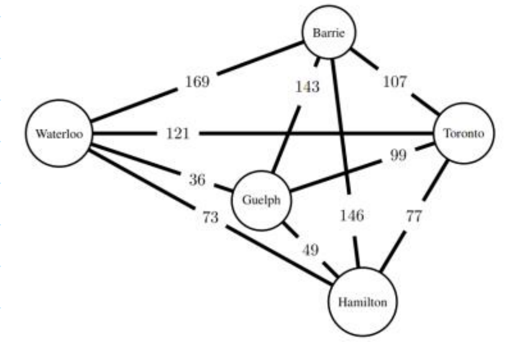
cost(p) = distance(WG) + distance(GT) + distance(TB)

= 36 + 99 + 107

= 242

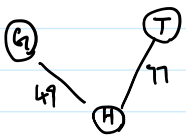
c) $h(n)$ = (min distance among all edges to the neighbours of node n) + (the total distance of the minimum spanning tree of all unvisited nodes) + (min distance from all unvisited cities to the start city)

Notice that the neighbours of node n are all other unvisited cities, and the minimum spanning tree of all other unvisited cities always exist. Both fact are true because the graph we have is a complete graph, so any two cities are connected. Finding the MST might be expensive, though.



Example: suppose the current state is (W, <W, B>), and we are looking for h(B). So there are three unvisited cities: G, T, and H

1. The closet unvisited city of B is T. So the minimum distance among all edges (BG, BT, BH) from B to all neighbors is 107.
2. All unvisited cities are G, T and H. It should be obvious that the MST of this subgraph is as follows:



   So the total distance of this MST is $49 + 77 = 126$
3. The edges from all other unvisited cities to the start city are: TW, GW, HW with distances 121, 36, 73, respectively. Therefore we choose the closet one, GW with distance 36.

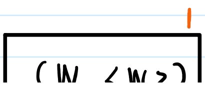
Therefore, $h(B) = 107 + 126 + 36 = 269$

d)

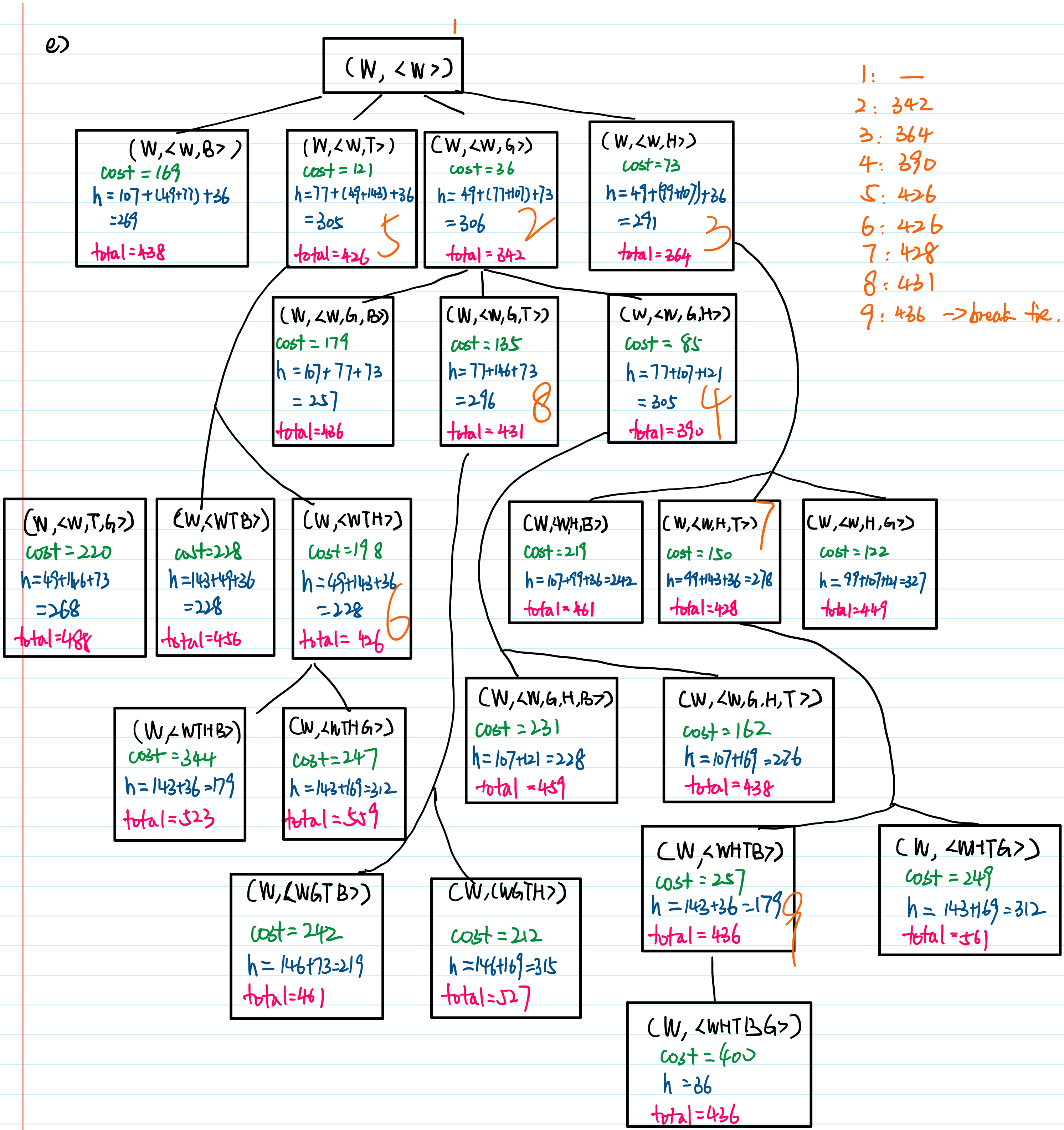
To prove, suppose that we have a complete graph $\mathrm{G(V, E)}$, where V is the set of all nodes and E is the set of all edges. For any edge $e \in E$, the distance/weight is given by $d(e)$, and suppose the start city given is a node $m \in V$.

Because of the graph is complete, let's assume that we are at a state $(m, < m, a, b >)$ WLOG. In other words, the current city we are at is b, and all unvisited cities are $E - \{m, a, b\}$. To achieve the goal state, we need to travel from b and visite all unvisited cities, and finally reach the start city m. The path $b, \ldots, m$ is made of three parts: $bx, x \ldots y, and\ ym$, where $x, y \in V$are arbitraty cities, again, since the graph is compelete.
By choosing the nearset unvisited nodes of b, the first part of our heuristic funciton is always less than or equal to $e(bx)$.
By choosing the nearest unvisited nodes of m, the third part of our heuristic function is always less than or equal to $e\left(ym\right)$
Then consider about the path from x to y that visits every unvisited cities. This path is a spanning tree of all unvisited cities, and the total weight is less than or equal to the total weight of the MST of these cities by definition. Therefore, the second part of our heuristic function is always less than or equal to the distance of path x...y.
Since each part of the sum in our heuristic function is less than the corresponding part of the actual cost, we can confirm that our heuristic function is admissible as it is a lower bound of the total cost to the goal state.

e)



e)



1: —
2: 342
3: 364
4: 390
5: 426
6: 426
7: 428
8: 431
9: 436 -> break tie.