CS 771

HW# Soln

(1) $f(n) = \omega\, g(n) + (8-\omega)\, h(n)$

(a) $\omega = 8 \Rightarrow f(n) = 8\, g(n)$. This is uniform cost search

(b) $\omega = 0 \Rightarrow f(n) = 8\, h(n)$. This greedy best first search

In both cases the constant factor 8 does not affect the solution

(c) $\omega = 4 \Rightarrow f(n) = 4\, g(n) + 4\, h(n) = 4\left(g(n) + h(n)\right)$

This is A^* search.

(d) $f(n) = \omega\, g(n) + (8-\omega)\, h(n) = \omega\left[g(n) + \left(\frac{8-\omega}{\omega}\right) h(n)\right]$

$= \omega\left[g(n) + \tilde{h}(n)\right]$

where $\tilde{h}(n) = \frac{8-\omega}{\omega}\, h(n) = \left(\frac{8}{\omega} - 1\right) h(n)$

Since h is admissible, for every node n, $h(n) \le h^*(n)$

If we can show that for every node n, $\tilde{h}(n) \le h(n)$, that would imply $\tilde{h}(n) \le h^*(n)$, leading to optimal search.

Therefore we seek to find a range of ω for which $\tilde{h}(n) \le h^*(n)$.

$\Rightarrow \left(\frac{8}{\omega} - 1\right) h(n) \le h(n) \Rightarrow \frac{8}{\omega} - 1 \le 1 \Rightarrow \frac{8}{\omega} \le 2 \Rightarrow \omega \ge 4$

Also note that we must have $\frac{8-\omega}{\omega} \ge 0$, otherwise $g(n) + \frac{8-\omega}{\omega} h(n) \le g(n)$.

$\frac{8-\omega}{\omega} \ge 0 \Rightarrow 8 - \omega \ge 0 \Rightarrow \omega \le 8$

$\therefore$ if we choose $8 \le \omega \le 8$ then the resulting search will be optimal.

~~Then can also do~~

$$\cancel{f(n) = (6-\omega)\left[\left(\frac{\omega}{6-\omega}\right) g(n) + h(n)\right]}$$

$\cancel{(6-\omega) > 0 \Rightarrow \omega < 6}$, ~~otherwise f(n) will be negative.~~

$$\cancel{\frac{\omega}{6-\omega} = 1 \Rightarrow \omega = 6-\omega \Rightarrow \omega = 3.}$$

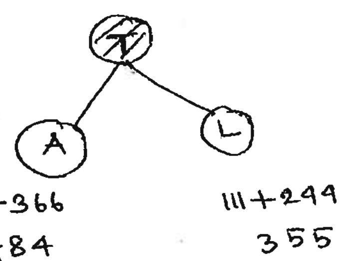
(2) We will use the first letter of each city to represent them.

● : visited.

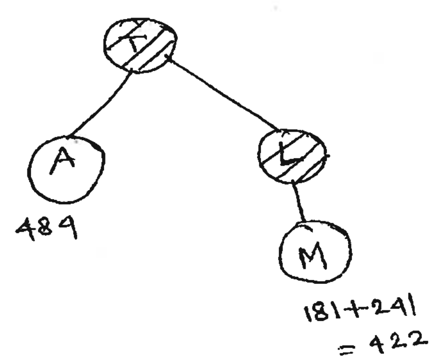
○ : frontier.

Step 1:

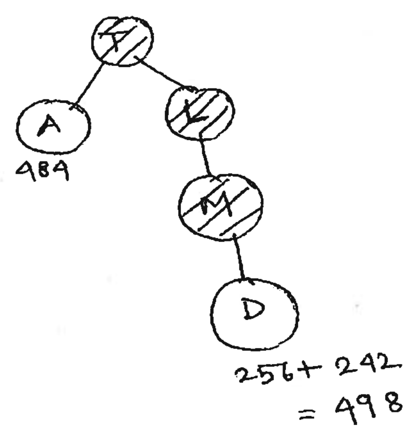
Step 2:



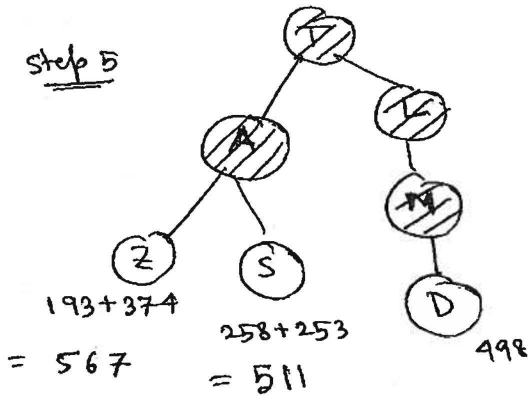
Step 3:



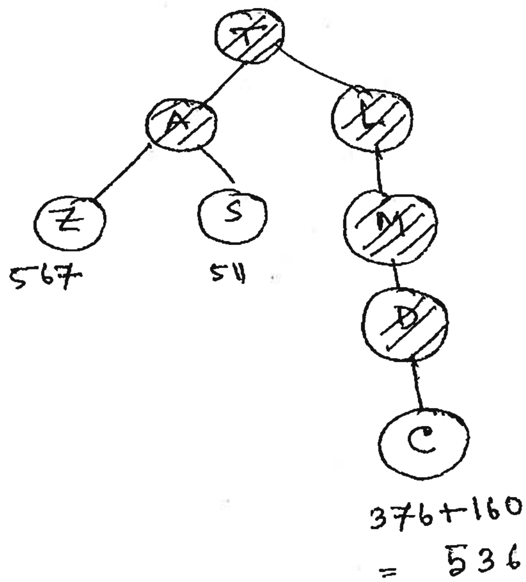
Step 4:



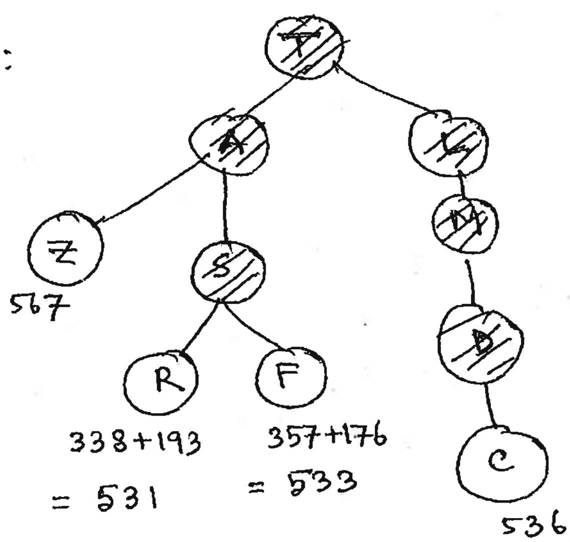
Step 5



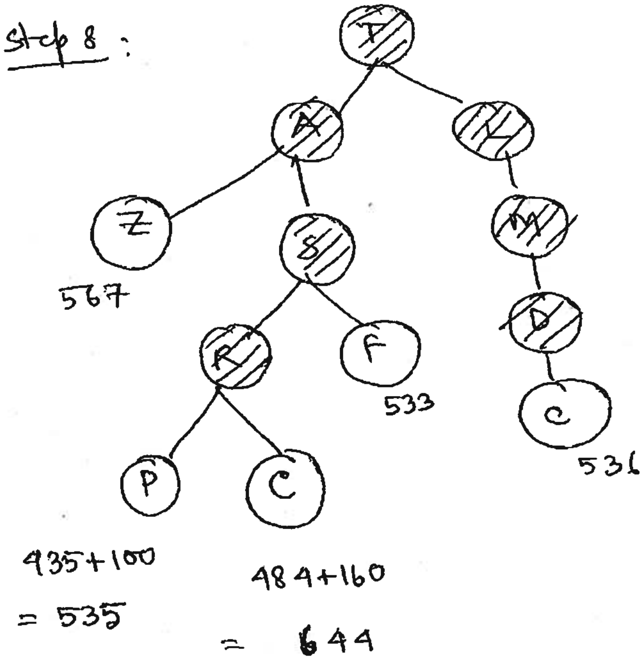
Step 6



step 7:



Step 8 :



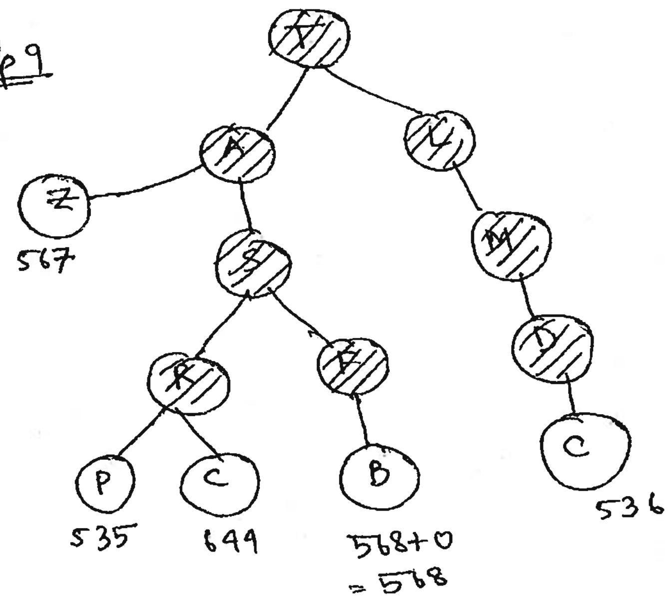
118
140
80
97

435

118
140
80
146

484

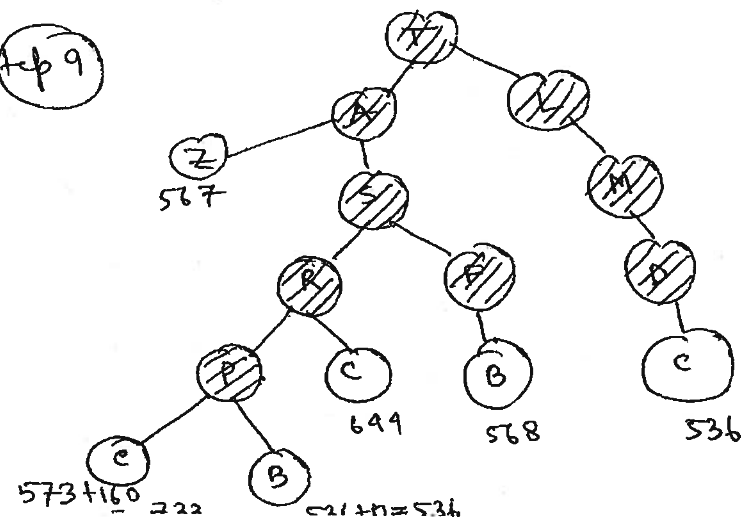
Step 9



118
140
99
211

568

Step 9



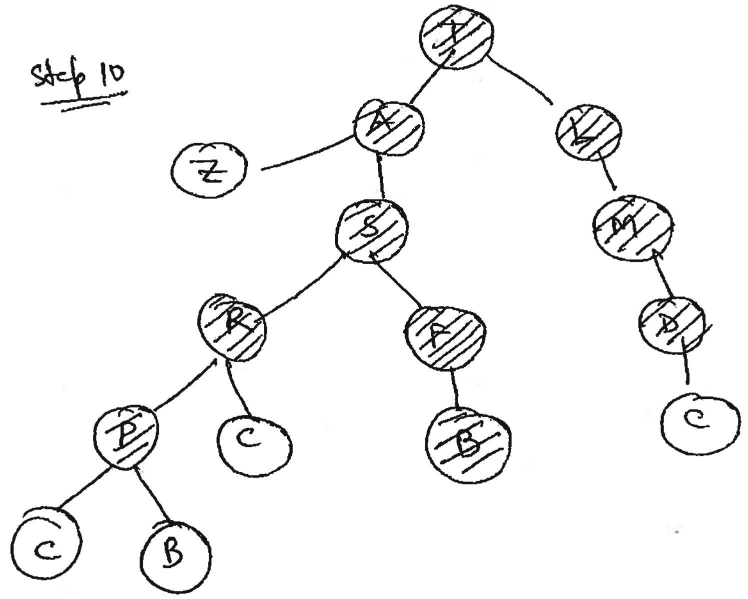
118
140
80
97
101

536

118
140
80
97
138

573

Step 10



(3) (a) If $T = \infty$, then it is a purely random walk.

Note that in simulated annealing next ~~a~~ node is set ~~to be~~ randomly as a successor of the current node. If it's value is higher than the value of current node then ~~e~~ this next node is selected. Otherwise (~~[illegible]~~ it's value is not higher than current node i.e. $\Delta E < 0$) then $e^{\Delta E/T} = 1$ as $T \to \infty$ and this node is selected with probability 1. That means no matter what, the randomly selected next node is selected. (See Algorithm of Simulated annealing from class slide, Ch 4).

(b) If $T = 0$, then it will not do any search and will always return the current node. (See Simmulated annealing algo from class slide)

(5) ~~[illegible]~~

④ (a) No. an optimal sear algorithm always finds the least cost soln. Since it finds a solution when one exists, it is complete.

(b) Yes. Completeness merely means the search algorithm will find a solution that solution may not be optimal or least cost soln.

(c) True. $f(n) = g(n) + h(n)$

If we ignore $h(n)$, then $f(n) = g(n)$, which is uniform cost search

" " " $g(n)$, " $f(n) = h(n)$, which is greedy best first search.

⑤(a) Breadth first Search (BFS)

—

(b) Depth first search. (DFS)

—