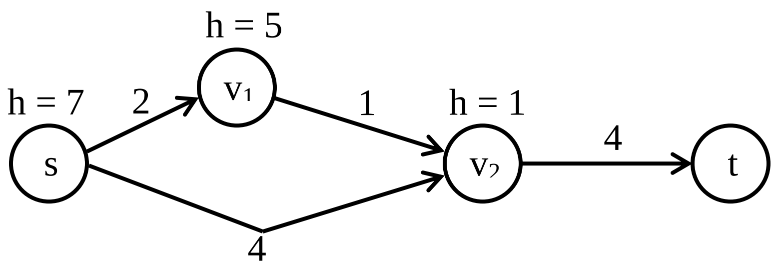
Supplementary Notes for Lecture 3 on Informed Search

Question 1:

What is a simple counter-example [1]that shows that only admissibility (i.e., but not consistency) will be sufficient for the (graph-based) A* search algorithm to produce an optimal solution?



We observe that h is admissible since it always under-estimates h*:

node n	h(n)	h*(n)
s	7	7 or 8
v_1	5	5
v_2	1	4
t	0	0

Note that h*(n) is the actual cost of reaching the goal from n; recall that for admissibility:

$$\forall n, h(n) \leq h^*(n)$$

However, in this case, we have:

node n child n'	h(n)	c(n, a, n') + h(n')
s, v_1	7	2 + 5 = 7
s, v_2	7	4 + 1 = 5
v_1, v_2	5	1 + 1 = 2
v_2, t	1	4 + 0 = 4

Notice that changing h(v_2) to 4 makes h consistent. As an exercise trace A* in this case.

For consistency, for every node n, and every successor of n (reached by a), we require:

$$h(n) \leq c(n, a, n') + h(n')$$

We observe that h is not consistent since the above is not the case between (i) s, v_2 and (ii) v_1, v_2.

A* (graph search) is applied as follows:

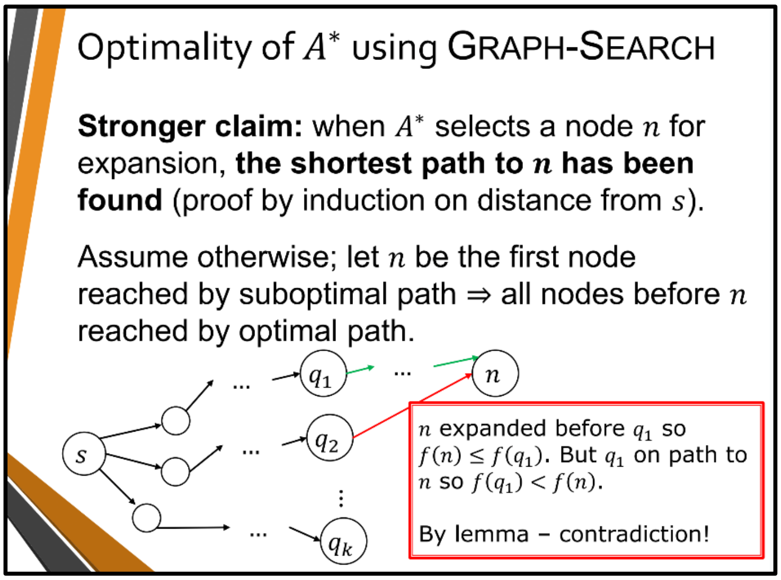
1. Add s to Frontier: [(s,7)], where 5 is calculated via: f(s) = g(s) + h(s) = 0 + 7
2. Pop Frontier; check goal test on s, and since False: expand node s
3. Add (v_1, 7) and (v_2, 5) to Frontier: [(v_2, 5), (v_1, 7)], as f(v_1) = 2 + 5 and f(v_2) = 4 + 1
4. Pop Frontier; check goal test on v_2, and since False: expand node v_2
5. Add (t, 8) to Frontier: [(v_1, 7), (t, 8)], as f(t) = 8 + 0
6. Pop Frontier; check goal test on v_1, and since False: expand node v_1

[1] Note that we are trying to find an example that shows that if we do NOT have a consistent h, then we may not find an optimal solution.

7. However, since v_2 has already been visited, we do not add it (graph search).
8. Pop Frontier; check goal test on t, and since True, solution is: s > v_2 > t (not optimal)

Question 2:

Note the following lecture slide. Why can't we have $f(q_1) = f(n)$? (And as such, no contradiction).



Assume that A* is following an optimal path until at some point, a node n is expanded, which is non-optimal. Let there be a node q_1 on the optimal path, such that n is expanded before q_1.

For the above to occur, n is expanded before q_1, and consequently, we must have $f(n) \leq f(q_1)$.

However, if q is on the path to n, then $f(q_1) < f(n)$.

The question above contemplates why, for the latter, we do not have $f(q_1) \leq f(n)$ instead.

The reason why we have $f(q_1) < f(n)$ is because if $f(n) = f(q_1)$, then n is on the optimal path (i.e., n is equally as good as q_1, and q_1 is on the optimal path), which is contrary to our original assumption.