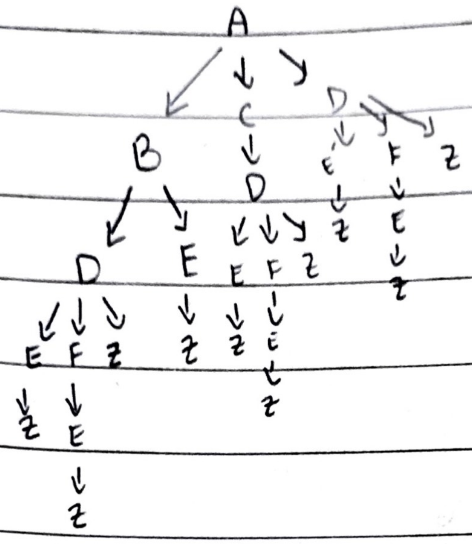
Date / /

Jonathan Tsu - HW1

Problem 1: Search Tree



Problem 2: UCT Search

A, C, D, F, E — Cost = 5

, D, F — Cost = 5

B — Cost: 6

, E — Cost: 7

, Z (done) — cost 7

Therefore, shortest path is A, C, D, F, E, Z

Problem 3: UCG Search

A, C, D, F, E, Z

Problem 4: Admissibility

Heuristic for D, h(D) must be in range of $0 \le h(D) \le 4$

↳ total cost ↳ 6 | +4 | 9
↳ 11 | +7 | 6
11 | +2 | 7

Problem 5: Consistency

Range of values for $h(D)$ for consistency are between $2 \leq h(D) \leq 3$

Problem 6: A^* Tree Search

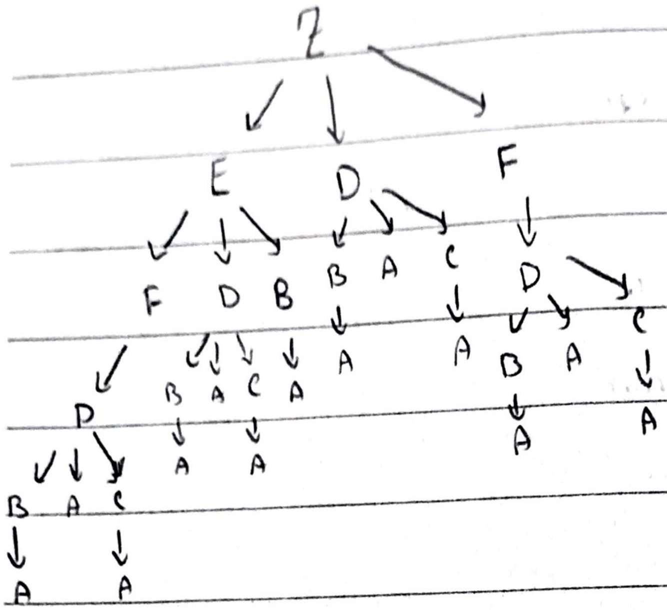
A, C, D, F, E, D, F, Z

Problem 7: A^* Graph Search

A, C, D, F, E, Z

Date / /

Problem 8: Reverse Search Tree



Problem 9: Reverse Uniform ~~Reverse~~ Tree Search

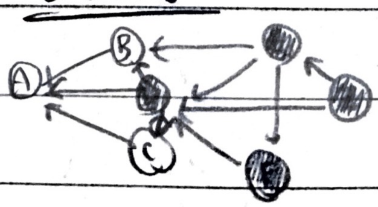
Z, E, F, D, C, B, A

Problem 10: Bidirectional UCT Search

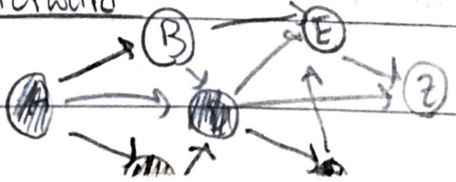
Forward:
A, C, D, F

Backward:
Z, E, F, D

Backward



Forward

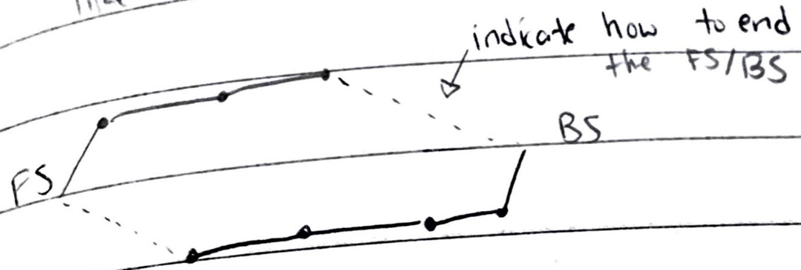


Converge on F & D

Date / /

Problem 11: Completeness of Bidirectional Search

Let us suppose Forward Search (FS) and Backwards Search (BS) are looking on a graph with the single goal state being where BS starts. Let's also say that the two have diverged like so:



Why can't this happen? If BS and FS are both choosing the lowest cost nodes, then as they approach their own specific ends they must be choosing the edge already chosen by the other FS/BS. That means they are both fact checks on each other and converge on the intersecting point, thus, it will be complete since they provide to each other a pathway to the ends.

Problem 12: Approximate Admissibility

Consider that $f(n) = g(n) + h(n)$

Then, looking at when we have a pessimistic heuristic, we convert this to,

$$f(n) = g(n) + h^*(n) + \varepsilon$$

Additionally, we also know that $h^*(n)$ is the shortest distance possible to the goal state, which correlates to the true cost to the optimal path to goal from $f(n)$. Therefore, we know that the portion of $f(n) = g(n) + h^*(n)$ is similar to when we observe via an admissible heuristic. Thus, by adding ε, we constrain our non-admissible's true cost (cost + heuristic) to be at most over the optimal by a value of ε. Note that as n approaches goal state, $h^*(n)$ falls by that similar amount, which is why we say that it is similar to admissible heuristic.

Date / /

Problem 13: Dominating Heuristics

We have that $h_1 > h_2$ for all n, but that both are under $h^*(n)$. This means that since they underestimate the true shortest path, they both are admissable. Then, we know that both will lead to the optimal solution.

Now, working at a state n, we know that A^* tree uses $f(n) = g(n) + h(n)$ where $g(n)$ is static but $h(n)$ will be either h_1 or h_2. Since we know that $h_1 > h_2$, then h_1 will choose more aggressively close to true cost paths. Since we know h_2 will also find the same path, it may look for less than optimal pathways. Additionally, this is tree search so nodes may be repeatedly observed and expanded. Therefore, because h_2 may observe more nodes than h_1 but still reach the same conclusion, h_1 is a more efficient search.