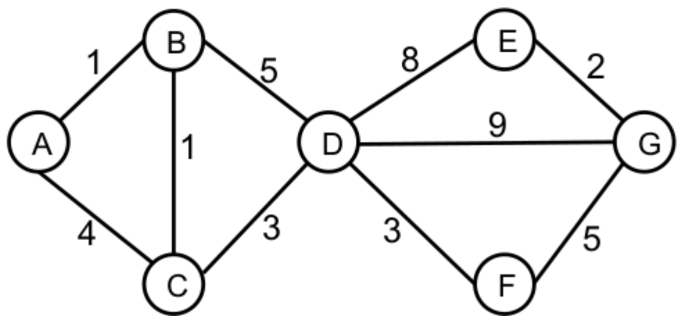
**CS 6300** **HW01: Search** **Ryan Dalby** **Due 25 January 2022**

Please use LaTeX to produce your writeups. See the Homework Assignments page on the class website for details.

# 1 Uninformed Search

Consider the state space graph shown below. A is the start state and G is the goal state. The costs for each edge are shown on the graph. Each edge can be traversed in both directions.




Execute the following search algorithms using priority queues, by filling in the search table for each part. (Not all steps will necessarily be used.)

Note that for each of the searches below nodes already included in the closed list (previously expanded nodes) were not added to the priority queue. Also note that alphabetical priority was enforced for tie breaks of depth or cost, where G had highest alphabetical priority.

1. **Breadth First Graph Search.**

| Step | Priority Queue | Expand |
|---|---|---|
| 1 | (A,1) | A |
| 2 | (A-B,2), (A-C,2) | B |
| 3 | (A-B-C,3), (A-B-D,3), (A-C,2) | C |
| 4 | (A-B-C,3), (A-B-D,3), (A-C-B,3), (A-C-D,3) | D |
| 5 | (A-B-D-E,4), (A-B-D-G,4), (A-B-D-F,4), (A-C-B,3), (A-C-D,3) | G |
| 6 | (A-B-D-G,4) is the solution | |

Note that for step 4 (A-B-C,3) was not expanded since C was on the closed list, it was subsequently dropped from the priority queue.

Note that for step 5 (A-C-B,3) and (A-C-D,3) were not expanded since B and D were on the closed list respectively, they were subsequently dropped from the priority queue.

2. **Depth First Graph Search.**

| **Step** | **Priority Queue** | **Expand** |
|---|---|---|
| 1 | (A,1) | A |
| 2 | (A-B,2), (A-C,2) | B |
| 3 | (A-B-C,3), (A-B-D,3), (A-C,2) | C |
| 4 | (A-B-C-D,4), (A-B-D,3), (A-C,2) | D |
| 5 | (A-B-C-D-E,5), (A-B-C-D-G,5), (A-B-C-D-F,5), (A-B-D,3), (A-C,2) | G |
| 6 | (A-B-C-D-G,5) is the solution | |

3. **Uniform Cost Graph Search.**

| **Step** | **Priority Queue** | **Expand** |
|---|---|---|
| 1 | (A,0) | A |
| 2 | (A-B,1), (A-C,4) | B |
| 3 | (A-B-C,2), (A-B-D,6), (A-C,4) | C |
| 4 | (A-B-C-D,5), (A-B-D,6), (A-C,4) | D |
| 5 | (A-B-C-D-E,13), (A-B-C-D-G,14), (A-B-C-D-F,8), (A-B-D,6) | F |
| 6 | (A-B-C-D-E,13), (A-B-C-D-G,14), (A-B-C-D-F-G,13) | G |
| 7 | (A-B-C-D-F-G,13) is the solution | |

Note that for step 4 (A-C,4) was not expanded since C was on the closed list, it was subsequently dropped from the priority queue.

Note that for step 5 (A-B-D,6) was not expanded since D was on the closed list, it was subsequently dropped from the priority queue.

## 2 Heuristic Search

1. Consider the two heurisitics $h_1$ and $h_2$, only one of which is consistent. Which one is consistent?

   $h_1$ **is** consistent because $\forall X, Y \quad h_1(X) - h_1(Y) \leq cost(X, Y)$ where $X$ and $Y$ are nodes in the state space graph.

   Note that $h_2$ is **not** consistent, a counter example that violates consistency is $h_1(B) - h_1(C) = 2 > cost(B, C) = 1$.

| Node | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| $h_1$ | 9.5 | 9 | 8 | 7 | 1.5 | 4 | 0 |
| $h_2$ | 10 | 12 | 10 | 8 | 1 | 4.5 | 0 |

2. Then do A* search with that heuristic.

| **Step** | **Priority Queue** | **Expand** |
|---|---|---|
| 1 | (A,9.5) | A |
| 2 | (A-B,10), (A-C,12) | B |
| 3 | (A-B-C,10), (A-B-D,13), (A-C,12) | C |
| 4 | (A-B-C-D,12), (A-B-D,13), (A-C,12) | D |
| 5 | (A-B-C-D-E,14.5), (A-B-C-D-G,14), (A-B-C-D-F,12), (A-B-D,13) | F |
| 6 | (A-B-C-D-E,14.5), (A-B-C-D-G,14), (A-B-C-D-F-G,13), (A-B-D,13) | |
| 7 | (A-B-C-D-F-G,13) is the solution | |

   Note that for step 4 (A-C,12) was not expanded since C was on the closed list, it was subsequently dropped from the priority queue.

   Note that for step 6 the tie-break of (A-B-C-D-F-G,13) and (A-B-D,13) went to (A-B-C-D-F-G,13) by tie-break rules before even checking the closed list.

3. Suppose you are completing the new heuristic function $h_3$ shown below. All the values are fixed except $h_3(B)$.

| Node | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| $h_3$ | 10 | ? | 9 | 7 | 1.5 | 4.5 | 0 |

   For each of the following conditions, write the set of values that are possible for $h_3(B)$. For example, to denote all non-negative numbers, write [0, $\infty$], to denote the empty set, write $\emptyset$, and so on.

   (a) What values of $h_3(B)$ make $h_3$ admissible?

$$h_3(B) = [0, 12]$$

   (b) What values of $h_3(B)$ make $h_3$ consistent?

$$h_3(B) = [9, 10]$$

(c) What values of $h_3(B)$ will cause A* graph search to expand from node A to C, then node A to B, then node A to B to D in that order?

$$h_3(B) = (12, 13)$$