A\* (A Star) Shortest Path

**Specification:**

Start with the given Java program “prog340”, which lets you select a file to read from your computer, reads the file, and interprets that file as the specification of a graph.[[1]](#footnote-1) Use A\* search as described in the notes and Poole & Mackworth textbook to find the shortest distance from a given start node to a single given goal node. Report the output as described below. Note that the test files will all be measuring distance between cities, so “cost” in some of the literature is “distance” in this specification.

**Input:**

I will test with two input files, the same twelve and forty-nine city files that were used for Deliverable C. distance between each pair of cities. These files are on the D2L site as file H12dist.txt and H49dist.txt.

The files list the actual and heuristic distances between pairs of cities. Notice that each graph is symmetric: the distance between two cities is the same in either direction. To get between cities without a direct connection, you need to go through other cities. There are two “special” cities in the D*nn*dist.txt files. Value “S” refers to the Start city for a search and value “G” refers to the goal city. *I will test by varying the Start and Goal cities.*

**Output:**

The output will be a set of lines showing the path that the algorithm expands at each step. When you discover the goal city, you are done, because the H*nn*dist.txt files show an admissible heuristic. Output to the console and to a file. The output should look like the SampleOutput section.

**Sample Output**

Most lines of output will show a path.

* The first set of paths is simply the 0-distance path from the start node to itself.
* The second set of paths adds paths to adjacent nodes by expanding the first path and removes the initial path.
* Later sets of paths choose the path with the lowest f-value and expand it, removing it from the path list.

On each line, show the whole path, the distance *d* from the start of the path to the end of the path, the heuristic estimate *h* from the end of the path to the goal, and the *f*-value ( *f = d + h* ) used to choose the next path to expand.

See the example below, followed by the table and map, then an annotated explanation of this output. The output on the rest of this page is the correct output for the shortest path from Boston to Dallas found using files D12dist.txt and H12dist.txt. Notice that we may find a shorter route to an intermediate node, so we calculate the distance to each intermediate node.

The output we are looking for is on the left, comments are on the right.

**Shortest Path from Boston (Bos) to Dallas (Dal)**

**PATH DIST HEUR F-VALUE**

**Bos 0 0 0**

**Bos-Chi 859 400 1259**

**Bos-Mpl 1127 400 1527**

**Bos-NY 207 700 907**

**Bos-NY-Chi 925**

**Bos-NY-Was 420 600 1020**

**Bos-NY-Was-Atl 958 400 1358**

**Bos-NY-Was-Chi 1001**

**Bos-Chi-Dal 1662**

***Do not enqueue, longer path to Chi***

***Do not enqueue, longer path to Chi***

***Found goal, BTW it’s OK if you fully expand the path Bos-Chi before finding Dal and terminating.***

**Commentary**

The table above does not indicate explicitly that when a path is expanded, the expanded path is removed from the path queue (and I would strongly suggest that you do this with a Priority Queue.

First, be sure that you understand the table above, and can reproduce it by hand using the files D12dist.txt and H12dist.txt. Also be sure that you can do Deliverable C (hopefully you see now why I asked you to do LCFS in such a convoluted way that I did, rather than in the simpler way that LCFS is done on the internet).

You will need to modify the Prog340.java file to read a second input file for the heuristics. You will also have some decisions to make about program architecture. Do you want the heuristic and the distances to be separate graphs, or do you want to add the heuristic distances as edges to the graph that you already have? Either way, remember that I regression test deliverables A, B, and C, so you don’t want to break this existing functionality.

Because I am not requiring a report with any of the programs this semester, the total points for programs are 235 not 255. So the course total will be 980 points, not 1000.

A map of united states with black circles

Description automatically generated

~ val Atl Bos Chi Dal Den LA Mia Mpl NY SF Sea Was

Atlanta ~ ~ ~ 581 725 ~ ~ 608 ~ ~ ~ ~ 538

Boston S ~ ~ 859 ~ ~ ~ ~ 1127 207 ~ ~ ~

Chicago ~ 581 859 ~ 803 928 ~ ~ 354 718 ~ ~ 601

Dallas G 725 ~ 803 ~ 667 1248 1116 ~ ~ ~ ~ ~

Denver ~ ~ ~ 928 667 ~ 812 ~ 697 ~ 957 1033 ~

LosAngeles ~ ~ ~ ~ 1248 812 ~ ~ ~ ~ 313 ~ ~

Miami ~ 608 ~ ~ 1116 ~ ~ ~ ~ ~ ~ ~ ~

Minneapolis ~ ~ 1127 354 ~ 697 ~ ~ ~ ~ ~ 1415 ~

NewYork ~ ~ 207 718 ~ ~ ~ ~ ~ ~ ~ ~ 213

SanFrancisco ~ ~ ~ ~ ~ 957 313 ~ ~ ~ ~ 691 ~

Seattle ~ ~ ~ ~ ~ 1033 ~ ~ 1415 ~ 691 ~ ~

Washington ~ 538 ~ 601 ~ ~ ~ ~ ~ 213 ~ ~ ~

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #1**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

***We start at Bos, the distance from Bos to itself is 0. So we “finish” Boston. Then we expand it by calculating the length of every path starting at Bos.***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #2**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

***We expand all paths from Bos. Our priority queue has Bos-NY 202 at the head. Since we haven’t finished New York yet, we finish it. Then we expand every path that starts Bos-NY that doesn’t go to a finished city. Note that the (degenerate) path Bos has been removed from the queue.***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #3**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

**Bos-NY-Chi 925**

**Bos-NY-Was 420**

**Washington 420 Bos-NY-Was**

***We expand all paths from Bos-NY. Our priority queue has Bos-Was 420 at the head. Since we haven’t finished Washington yet, we finish it. Then we expand every path that starts Bos-NY-Was that doesn’t go to a finished city.***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #4**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

**Bos-NY-Chi 925**

**Bos-NY-Was 420**

**Washington 420 Bos-NY-Was**

**Bos-NY-Was-Chi 1021**

**Bos-NY-Was-Atl 958**

**Chicago 859 Bos-Chi**

***We expand all paths from Bos-NY-Was. Our priority queue has Bos-Chi 859 at the head. Since we haven’t finished Chicago yet, we finish it. Then we expand every path that starts Bos-Chi that doesn’t go to a finished city.***

***Note: We will never expand another path ending in Chi, in particular we will never expand Bos-NY-Chi. The best way to handle this is to delete it when it becomes the shortest path on the priority queue.***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #5**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

**Bos-NY-Chi 925**

**Bos-NY-Was 420**

**Washington 420 Bos-NY-Was**

**Bos-NY-Was-Chi 1021**

**Bos-NY-Was-Atl 958**

**Chicago 859 Bos-Chi**

**Bos-Chi-Mpl 1213**

**Bos-Chi-Den 1787**

**Bos-Chi-Dal 1662**

**Bos-Chi-Atl 1441**

**Atlanta 958 Bos-NY-Was-Atl**

***We expand all paths from Bos-Chi. Our priority queue has Bos-NY-Chi 925 at the head. Since we have already finished Chi we remove it. The next shortest path is Bos-NY-Was-Atl 958. That’s the one we expand, finishing Atlanta in the process. we expand every path that starts Bos-NY-Was-Atl that doesn’t go to a finished city.***

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #6**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

**Bos-NY-Chi 925**

**Bos-NY-Was 420**

**Washington 420 Bos-NY-Was**

**Bos-NY-Was-Chi 1021**

**Bos-NY-Was-Atl 958**

**Chicago 859 Bos-Chi**

**Bos-Chi-Mpl 1213**

**Bos-Chi-Den 1787**

**Bos-Chi-Dal 1662**

**Bos-Chi-Atl 1441**

**Atlanta 958 Bos-NY-Was-Atl**

**Bos-NY-Was-Atl-Dal 1683**

**Bos-NY-Was-Atl-Mia 1566**

**Minneapolis 1127 Bos-Mpl**

*The shortest remaining path, at the head of the queue, Bos-NY-Was-Chi 1021, ends in a city already finished so we discard it. This leaves us with Bos-Mpl 1127 at the head of the queue. We finish Minneapolis and expand Bos-Mpl*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #7**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

**Bos-NY-Chi 925**

**Bos-NY-Was 420**

**Washington 420 Bos-NY-Was**

**Bos-NY-Was-Chi 1021**

**Bos-NY-Was-Atl 958**

**Chicago 859 Bos-Chi**

**Bos-Chi-Mpl 1213**

**Bos-Chi-Den 1787**

**Bos-Chi-Dal 1662**

**Bos-Chi-Atl 1441**

**Atlanta 958 Bos-NY-Was-Atl**

**Bos-NY-Was-Atl-Dal 1683**

**Bos-NY-Was-Atl-Mia 1566**

**Minneapolis 1127 Bos-Mpl**

**Bos-Mpl-Den 1824**

**Bos-Mpl-Sea 2542**

**Miami 1566 Bos-NY-Was-Atl-Mia**

*The shortest remaining path, at the head of the queue, Bos-Chi-Mpl 1213, ends in a city already finished so we discard it. He next shortest path, Bos-Chi-Atl 1441 is also discarded since Atl is finished. This leaves us with Bos-NY-Was-Atl-Mia 1566 at the head of the queue. We finish Miami and expand Bos-NY-Was-Atl-Mia*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shortest Path from Boston (Bos) to Dallas (Dal) Step #8**

**PATH DIST CITY MIN\_DIST PATH**

**Bos 0**

**Boston 0 Bos**

**Bos-NY 202**

**Bos-Chi 859**

**Bos-Mpl 1127**

**NewYork 202 Bos-NY**

**Bos-NY-Chi 925**

**Bos-NY-Was 420**

**Washington 420 Bos-NY-Was**

**Bos-NY-Was-Chi 1021**

**Bos-NY-Was-Atl 958**

**Chicago 859 Bos-Chi**

**Bos-Chi-Mpl 1213**

**Bos-Chi-Den 1787**

**Bos-Chi-Dal 1662**

**Bos-Chi-Atl 1441**

**Atlanta 958 Bos-NY-Was-Atl**

**Bos-NY-Was-Atl-Dal 1683**

**Bos-NY-Was-Atl-Mia 1566**

**Minneapolis 1127 Bos-Mpl**

**Bos-Mpl-Den 1824**

**Bos-Mpl-Sea 2542**

**Miami 1566 Bos-NY-Was-Atl-Mia**

**Bos-NY-Was-Atl-Mia-Dal 2682**

**Dallas 1662 Bos-Chi-Dal**

*The shortest remaining path, at the head of the queue, Bos-Chi-Dal 1662. We finish Dallas. Since this is our goal node we are done.*

1. See the “prog340.doc” file for details [↑](#footnote-ref-1)