Extract either archive attached and put the assign3testing directory in the same directory as your project.

Then either copy and paste into your terminal, makefile, or execute with a bash script the following code:

```bash
./dijkstra assign3testing/graph_assign3example_1.txt > assign3testing/myoutput/graph_assign3example_1.txt
./dijkstra assign3testing/graph_assign3example_2.txt > assign3testing/myoutput/graph_assign3example_2.txt
./dijkstra assign3testing/graph_whitespace.txt > assign3testing/myoutput/graph_whitespace.txt
./dijkstra assign3testing/graph_empty.txt > assign3testing/myoutput/graph_empty.txt
./dijkstra assign3testing/graph_single.txt > assign3testing/myoutput/graph_single.txt
./dijkstra assign3testing/graph_multiplepaths_1.txt > assign3testing/myoutput/graph_multiplepaths_1.txt
./dijkstra assign3testing/graph_multiplepaths_2.txt > assign3testing/myoutput/graph_multiplepaths_2.txt
./dijkstra assign3testing/graph_multiplevertexdefs.txt > assign3testing/myoutput/graph_multiplevertexdefs.txt
./dijkstra assign3testing/graph_sparce_1.txt > assign3testing/myoutput/graph_sparce_1.txt
./dijkstra assign3testing/graph_sparce_2.txt > assign3testing/myoutput/graph_sparce_2.txt
./dijkstra assign3testing/graph_random_10.txt > assign3testing/myoutput/graph_random_10.txt
./dijkstra assign3testing/graph_random_50.txt > assign3testing/myoutput/graph_random_50.txt
./dijkstra assign3testing/graph_random_200.txt > assign3testing/myoutput/graph_random_200.txt
diff assign3testing/myoutput/graph_assign3example_1.txt assign3testing/output/graph_assign3example_1.txt
diff assign3testing/myoutput/graph_assign3example_2.txt assign3testing/output/graph_assign3example_2.txt
diff assign3testing/myoutput/graph_whitespace.txt assign3testing/output/graph_whitespace.txt
diff assign3testing/myoutput/graph_empty.txt assign3testing/output/graph_empty.txt
diff assign3testing/myoutput/graph_single.txt assign3testing/output/graph_single.txt
diff assign3testing/myoutput/graph_multiplepaths_1.txt
```
If it looks like nothing happened, we match.

**File Attachments**
1) `assign3testing.zip`, downloaded 42 times

---

**Subject: Re: Most Okayest Testing Files Ever**
Posted by *davidmccoy* on Wed, 22 Mar 2017 22:39:29 GMT

Nice speed on getting this done! B/c of the hackathon, I wouldn't have gotten testing files out until next Tuesday, so good to see some others tests out there to use. I hope to get this proj. done soon and corroborate your results.

---

**Subject: Re: Most Okayest Testing Files Ever**
Posted by *jarobinson3* on Sun, 26 Mar 2017 23:21:37 GMT

for graph_random_10.txt you output

```
0 : 0
----
0 : 1
1 : 6(1)1 2(0)3 7(1)3 9(0)4
2 : 3(6)4 5(6)5 4(7)8 8(7)8
----
```

The graph is undirected so I am not sure how you got your output stating that vertex 0 and the remaining vertices are disconnected. In your output you even say vertex 2 comes from vertex 0!
jarobinson3 wrote on Sun, 26 March 2017 18:21 for graph_random_10.txt you output

```
0 : 0
----
0 : 1
 1 : 6(1)1 2(0)3 7(1)3 9(0)4
 2 : 3(6)4 5(6)5 4(7)8 8(7)8
----
```

The graph is undirected so I am not sure how you got your output stating that vertex 0 and the remaining vertices are disconnected. In your output you even say vertex 2 comes from vertex 0!

Should be fixed now. The issue involved my heap not updating the extreme pointer during particular calls to decreaseKey.

-

Now I get a difference for graph_random_50.txt and graph_random_200.txt

It seems I get the same distances but a different path for node 39.

For instance to get to node 39 you do: 0->45->25->11->39, but I do 0->27->34->39

I think something is wrong with your bfs (unless there is something I missed in the assignment).

Here is my output for graph_random_50.txt

```
Mine

0 : 0
 1 : 43(0)1 3(0)2 2(0)3 45(0)3 18(0)4 27(0)5 28(0)10
 2 : 29(43)3 24(45)4 25(45)4 4(43)5 8(3)5 31(45)5 33(3)6 34(27)6 36(3)6 48(3)6 41(43)9
 3 : 21(29)4 19(24)5 46(29)5 38(4)6 40(29)6 6(4)7 11(25)7 12(36)7 16(34)7 44(33)7 47(24)7
 39(34)8 5(25)9 10(48)9 22(25)9 37(33)9 30(8)10 32(4)10 35(48)12
 4 : 20(21)5 7(19)6 9(46)6 1(46)7 17(19)7 13(16)8 26(16)8 23(11)9 49(21)9 15(6)11
 5 : 14(9)9 42(17)12
----
```
Subject: Re: Most Okayest Testing Files Ever
Posted by SSinischo on Mon, 27 Mar 2017 23:06:39 GMT

I had understood the specs such that, when given the choice between two paths of equal distance, the parent vertex with the smallest value will be chosen for the path.

Is this not appropriate behavior? I suppose it makes sense to prioritize a lesser amount of steps...

Subject: Re: Most Okayest Testing Files Ever
Posted by SSinischo on Mon, 27 Mar 2017 23:27:06 GMT

Fixed.

Subject: Re: Most Okayest Testing Files Ever
Posted by lusth on Tue, 28 Mar 2017 12:12:01 GMT

The graph graph_random_50.txt has been added to the test3 dropbox testing procedure (with self loops removed).

Subject: Re: Most Okayest Testing Files Ever
Posted by SSinischo on Tue, 28 Mar 2017 14:51:09 GMT
lusth wrote on Tue, 28 March 2017 07:12

The graph graph_random_50.txt has been added to the test3 dropbox testing procedure (with self loops removed).

There is another issue. Node 41 has two paths with distance 9 and is 2 steps away from the source. My algorithm selects the path that gives 41 the lowest parent value, which is 3 in my case. Yours selects 43 for the parent. Which is correct, and if yours is exhibiting appropriate behavior, what is the criteria for determining the parent when such a situation arises?

Subject: Re: Most Okayest Testing Files Ever
Post by lusth on Tue, 28 Mar 2017 21:04:47 GMT
View Forum Message <> Reply to Message

The tie breaking rule is if there are multiple candidates to be added to the shortest path tree, the one with the smallest vertex number wins. This translates to the vertex comparator, in that if key values are the same, then the vertex with the smaller number is "smaller".

Also, once a predecessor has been set, it is never reset unless there is a shorter path.

Subject: Re: Most Okayest Testing Files Ever
Post by nboltralik on Thu, 30 Mar 2017 17:24:00 GMT
View Forum Message <> Reply to Message

For the fifth file, graph_single.txt, there is no space between the last number and the semicolon and it seems like there should be according to this post http://beastie.cs.ua.edu/forums/index.php?t=msg&th=1740&goto=7942&amp;msg_7942

Subject: Re: Most Okayest Testing Files Ever
Post by lusth on Fri, 31 Mar 2017 13:41:38 GMT
View Forum Message <> Reply to Message

There should always be whitespace before and after a semicolon.

Subject: Re: Most Okayest Testing Files Ever
Post by lusth on Fri, 31 Mar 2017 13:42:34 GMT
View Forum Message <> Reply to Message

Again, here are the rules for predecessors:

According to dijkstra's algorithm in the book, the only time a predecessor changes is when the key value can be made smaller. The only tie breaker in the spec concerning the algorithm is when there are competing values (all with the same key) vying to become the extreme value in the priority queue. The tie breaker in that case is the vertex number.