Computer programming

Just another programming weblog

2007-11-15

All Sources Shortest Path: The Floyd-Warshall Algorithm

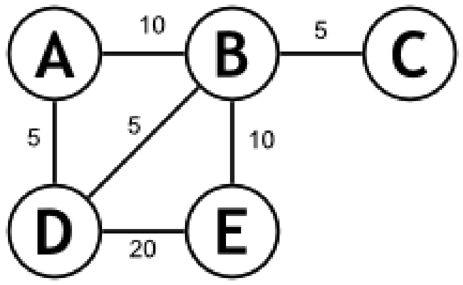
Posted by scvalex under Algorithms, Graphs | Tags: algorithm, C, explanation, floyd-warshall, graph, shortest path, sourcecode, tutorial |
[68] Comments

In this article I describe the Floyd-Warshall algorithm for finding the shortest path between all nodes in a graph. I give an informal proof and provide an implementation in C.

Shortest paths

The **shortest path** between two nodes of a graph is a sequence of connected nodes so that the sum of the edges that inter-connect them is minimal.

Take this graph,



There are several paths between *A* and *E*:

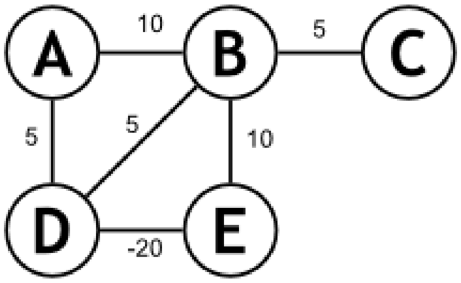
Path 1: *A -> B -> E 20*
Path 2: *A -> D -> E 25*
Path 3: *A -> B -> D -> E 35*
Path 4: *A -> D -> B -> E 20*

There are several things to notice here:

1. There can be more then one route between two nodes
2. The number of nodes in the route isn't important (**Path 4** has *4* nodes but is shorter than **Path 2**, which has *3* nodes)
3. There can be more than one path of minimal length

Something else that should be obvious from the graph is that any path worth considering is simple. That is, you only go through each node **once**.

Unfortunately, this is not always the case. The problem appears when you allow negative weight edges. This isn't by itself bad. But if **a loop of negative weight appears**, then **there is no shortest path**. Look at this example:



Look at the path *B -> E -> D -> B*. This is a loop, because the starting node is the also the end. What's the cost? It's *10 – 20 + 5 = -5*. This means that adding this loop to a path once lowers the cost of the path by *5*. Adding it twice would lower the cost by *2 * 5 = 10*. So, whatever shortest path you may have come up with, you can make it smaller by going through the loop one more time. BTW there's no problem with a negative cost path.

The Floyd-Warshall Algorithm

This algorithm calculates the length of the shortest path between all nodes of a graph in O(V^3) time. Note that it doesn't actually find the paths, only their lengths.

Let's say you have the adjacency matrix of a graph. Assuming no loop of negative values, at this point you have the minimum distance between any two nodes which are connected by an edge.

```
A B C D E
A 0 10 0 5 0
B 10 0 5 5 10
C 0 5 0 0 0
D 5 5 0 0 20
E 0 10 0 20 0
```

The graph is the one shown above (the first one).

The idea is to try to interspace *A* between any two nodes in hopes of finding a shorter path.

```
A B C D E
A 0 10 0 5 0
B 10 0 5 5 10
```

```
C 0 5 0 0 0
D 5 5 0 0 20
E 0 10 0 20 0
```

Then try to interspace *B* between any two nodes:

```
A B C D E
A 0 10 15 5 20
B 10 0 5 5 10
C 15 5 0 10 15
D 5 5 10 0 15
E 20 10 15 15 0
```

Do the same for *C*:

```
A B C D E
A 0 10 15 5 20
B 10 0 5 5 10
C 15 5 0 10 15
D 5 5 10 0 15
E 20 10 15 15 0
```

Do the same for *D*:

```
A B C D E
A 0 10 15 5 20
B 10 0 5 5 10
C 15 5 0 10 15
D 5 5 10 0 15
E 20 10 15 15 0
```

And for *E*:

```
A B C D E
A 0 10 15 5 20
B 10 0 5 5 10
C 15 5 0 10 15
D 5 5 10 0 15
E 20 10 15 15 0
```

This is the actual algorithm:

```
# dist(i,j) is "best" distance so far from vertex i to vertex j

# Start with all single edge paths.
 For i = 1 to n do
     For j = 1 to n do
         dist(i,j) = weight(i,j)

 For k = 1 to n do # k is the `intermediate' vertex
     For i = 1 to n do
         For j = 1 to n do
             if (dist(i,k) + dist(k,j) < dist(i,j)) then # shorter path?
                 dist(i,j) = dist(i,k) + dist(k,j)
```

The Programme

Here's the code in C(floyd_warshall.c):

```c
#include <stdio.h>

int n; /* Then number of nodes */
int dist[16][16]; /* dist[i][j] is the length of the edge between i and
            it exists, or 0 if it does not */

void printDist() {
    int i, j;
    printf("    ");
    for (i = 0; i < n; ++i)
        printf("%4c", 'A' + i);
    printf("\n");
    for (i = 0; i < n; ++i) {
        printf("%4c", 'A' + i);
        for (j = 0; j < n; ++j)
            printf("%4d", dist[i][j]);
        printf("\n");
    }
    printf("\n");
}

/*
    floyd_warshall()

    after calling this function dist[i][j] will the the minimum distance
    between i and j if it exists (i.e. if there's a path between i and j
    or 0, otherwise
*/
void floyd_warshall() {
    int i, j, k;
    for (k = 0; k < n; ++k) {
        printDist();
33            for (i = 0; i < n; ++i)
34                for (j = 0; j < n; ++j)
35                    /* If i and j are different nodes and if
36                        the paths between i and k and between
37                        k and j exist, do */
38                    if ((dist[i][k] * dist[k][j] != 0) && (i != j))
39                        /* See if you can't get a shorter path
40                            between i and j by interspacing
41                            k somewhere along the current
42                            path */
43                        if ((dist[i][k] + dist[k][j] < dist[i][j]) ||
44                            (dist[i][j] == 0))
45                            dist[i][j] = dist[i][k] + dist[k][j];
46        }
47        printDist();
48    }
49
50    int main(int argc, char *argv[]) {
51        FILE *fin = fopen("dist.txt", "r");
52        fscanf(fin, "%d", &n);
53        int i, j;
54        for (i = 0; i < n; ++i)
55            for (j = 0; j < n; ++j)
56                fscanf(fin, "%d", &dist[i][j]);
57        fclose(fin);
58
59        floyd_warshall();
60
61        return 0;
62    }
```

Note that of the above programme, all the work is done by only five lines (30-48).

That's it. Good luck. Always open to comments.

68 Responses to "All Sources Shortest Path: The Floyd-Warshall Algorithm"

1. Flavio Says:

   2007-12-06 at 20:47
   Very good algorithm ;D

   Helped a lot. Thanks!

   Reply

2. Danielle Says:

   2008-04-22 at 2:28
   I found this helpfull although i still dont understand the formular that i keep hearing about in other places online, maybe you should include that somewhere, so we all have a better understanding.

   Reply
3. rocker Says:

   2008-08-11 at 18:17
   really very helpful at least for beginners.

   Reply
4. Tarif Ezaz Says:

   2008-09-26 at 11:48
   Learned it a week ago. Thanks for the recap.

   Regards
   Tarif

   Reply
5. mh Says:

   2008-10-11 at 9:15
   does this algo depend on the density of the graph.
   from my understanding it should not as we are using an adjacency matrix and a sparse graph and dense graph need the same amount of work.

   please comment….

   Reply
6. rapist Says:

   2008-11-30 at 13:43
   excellent helped a lot

   Reply
7. fuker Says:

   2008-11-30 at 13:45
   if i had read dis before i would have cracked this xam…but am unlukky

   Reply
8. ja bwoi Says:

2008-12-11 at 9:22
THANX PLENTY

Reply

9. Definitely love msn, fine website. Have a nice day. Says:

   2008-12-22 at 12:45
   Shawn S…

   Kudos, just stopped by….

   Reply

10. alhgamy Says:

    2009-01-27 at 22:36
    thanxxxxxxxxxxx alot

    Reply

11. rajesh Says:

    2009-02-15 at 7:01
    very very goooooooooooooooooooooooooooooooood!!!!!!!!!!!!

    Reply

12. nikos Says:

    2009-03-22 at 14:30
    Very good algorithm. What about if we want to know not only the weight of the shortest path but the shortest path itself.
    Thanks Nikos

    Reply

    1. Max Says:

       2010-06-02 at 5:01
       I also have the same question as Miko :

       What about if we want to know not only the weight of the shortest path but the shortest path itself.
       Thanks Nikos.

       I'll explain the problem i'm using Floyd for :

       I have to open a text file with a bunch of 4 letters words, and the user has to pick 2 of those words, then my program needs to find the shortest way of going from word1 to word2 by only changing one letter at a time (ie. bear, beer, beez, …)

       What I did is make a matrix with all the words. put 1 on (i,j) if the word i can be transformed into j with only 1 letter modification,

My problem is that I don’t need the number of words used to go from word1 to word2 but the list of word itself… can the floyd algorithm do that ? or should I use dijkstra ?

Thanks in advance,
Max

Reply

1. scsum Says:

   2013-04-20 at 4:26
   Look up Edit Distance. That’s what you’re looking for, it also involves dynamic programming.

13. harro Says:

    2009-03-24 at 18:25
    what would a sample input file look like for this program?

    Reply

    1. Opu Says:

       2009-07-16 at 0:55
       The main term is how you fill the distance table.(usually by taking input 1,2,4.. like this). Rather then this You need to allocate the distance from one node to every node in the table.(here use a default value).

       Reply

14. Opu Says:

    2009-07-16 at 0:51
    Oh.. I don’t know how to thank you.
    This discussion help me to learn the algorithm. And i solved a program using Floyd in the same time!!! :D

    Reply

15. prasanth Says:

    2009-08-10 at 15:05
    thankyou for giving this code.actually when i saw the psudo code i understand it with some confusion .but when i saw the source code it was cleared . thanks a lot

    Reply

16. Lalith Says:

    2009-09-08 at 3:00

I too have same doubt. How do we construct the shortest path out of the final adjacency matrix is obtained?

Reply

17. King Says:

    2009-10-05 at 17:12
    Thanx ……
    It's Very helpful………..

    Reply

18. Virtualized Says:

    2009-10-08 at 13:35
    Hi

    It is a well written and well explained code. It can be compiled and built without any problem.

    I have executed it and also created a file dist.txt(which contains the adjacency matrix of my graph).

    But it does not seem to read that file upon execution and does not give any results(all pairs shortest paths).

    Can anybody please throw some light on this issue. It would be a great help and highly appreciated.

    Thanking You

    Best Regards

    Reply

    1. sc Says:

       2010-04-14 at 14:04
       Hi

       Can you let me know how did u give the values in the dist.txt ?

       Thank you
       sc

       Reply

    2. sc Says:

       2010-04-14 at 14:21

sorry..i think i figured out the dist.txt..
thanks anyways.
sc

Reply

1. ayush Says:

   2010-05-07 at 18:22
   hey could you help me out with dist.txt file..what should i write out in that

19. pedram Says:

    2009-10-08 at 14:21
    thanks man

    Reply

    1. Virtualized Says:

       2009-10-08 at 14:45
       Hi

       Can you please reply to my question posted above?

       Thanking You

       Best Regards

       Reply

20. ritesh Says:

    2009-11-18 at 9:35
    gud…needs to be bit more interactively depicted and elaborated

    Reply

21. dall Says:

    2009-11-23 at 16:56
    Many of life's failures are people who did not realize how close they were to success when they gave up.

    Reply

22. jack Says:

    2010-01-04 at 13:36
    it was nice n i really got help from it, thanks to it that it helped me in my semester paper…

    Reply

23. aa Says:

2010-03-04 at 7:05
need a java version

Reply

24. Ondrej Says:

    2010-03-21 at 19:29
    Does anybody have source code of parallel version of this algorithm (MPI)? If yes, please contact me on mail. Thx.

    Reply

25. vivek Says:

    2010-04-22 at 11:09
    thnx a lot

    Reply

26. Istiaque Says:

    2010-04-22 at 18:46
    Does anybody can say how the following line will work.

    FILE *fin = fopen("dist.txt", "r");

    I know I have to made a file. but in that file what will be stay?

    Reply

27. datanium Says:

    2010-05-04 at 3:38
    thank you very much
    soo soo helpfull

    Reply

28. ayush Says:

    2010-05-07 at 18:21
    please tell me what should i write in dist.txt..i.e in which format..

    Reply

29. saly Says:

    2010-05-11 at 21:53
    plesa I need code for calculation between each pairs of nodes by dijkestra algorithm

    pleassssssss

Reply

30. saly Says:

    2010-05-11 at 21:54

    plesa I need code for calculation shortest path between each pairs of nodes by dijkestra algorithm

    pleasssssssss

    Reply

31. Max Says:

    2010-06-02 at 5:02

    I also have the same question as Miko :

    What about if we want to know not only the weight of the shortest path but the shortest path itself.
    Thanks Nikos.

    I'll explain the problem i'm using Floyd for :

    I have to open a text file with a bunch of 4 letters words, and the user has to pick 2 of those words, then my program needs to find the shortest way of going from word1 to word2 by only changing one letter at a time (ie. bear, beer, beez, …)

    What I did is make a matrix with all the words. put 1 on (i,j) if the word i can be transformed into j with only 1 letter modification,

    My problem is that I don't need the number of words used to go from word1 to word2 but the list of word itself… can the floyd algorithm do that ? or should I use dijkstra ?

    Thanks in advance,
    Max

    Reply

32. lalithmaddali Says:

    2010-06-02 at 5:53

    Hello guys,
    Ah .. I used this thread to implement the algorithm long back. I remember printing shortest path in the following way. It will be straight forward but if need some code I can do it when I find free time.

    1) Maintain an n x n Path matrix along with dist matrix.

    2) Update path matrix in like this:
    When ever the value of particular cell in distance matrix is changed because a new shortest distance was found update corresponding cell in Path matrix with the node joining (here it is

k). Lot of rewriting can happen on cells but it will be useful to retrieve information of shortest path.

3) Now that we have the path array. Shortest path between i, j can be found in following way. The cell in i,j gives the node k which is closest to j and in the shortest path connecting i and j. Thus k is last but one node in our path. Now do the same with i, k and so on work backwards until u reach i, i cell. Finally reversing the cell values you get the actual shortest path.

Thanks,
Lalith

Reply

1. Max Says:

   2010-06-02 at 6:45
   Thanks for your post lalith, glad to see there is a way to do it. So I would need to create an extra array to keep the path along the way ? Seems logical !
   Where in the floyd_warshall function should I update this array ?

   I'll try some more of what you said and hopefully I can make it work soon !

   Here is a part of my code for the assignment I got to do in C. (see below)

   Thanks

```
int path[size][size];

//void floyd_warshall()
for (k = 0; k < size; ++k) {
for (i = 0; i < size; ++i)
for (j = 0; j < size; ++j)
if ((dist[i][k] * dist[k][j] != 0) && (i != j))
if ((dist[i][k] + dist[k][j] < dist[i][j]) ||
(dist[i][j] == 0))
{

dist[i][j] = dist[i][k] + dist[k][j];
path[i][j] = k;
}
}

//display the dist table (test)
for (c=0; c<size; c++)
{
for (d = 0; d<size; d++)
printf("%d|", dist[c][d]);
printf("\n");
}
```

Reply

33. lalithmaddali Says:

    2010-06-02 at 14:13
    Hi Max,
    Yes your path array is right. Now the trick is to print the path properly. Follow my instructions above to print out path between two nodes.

    pseudo code should be like this:

```
function printPath(path, i, j):
shortest_path = [] //(new array)
while true:
k = path[i][j]
shortest_path += k
j = k
if i == j:
break
// now we have shortest_path but in reverse order so reverse it to get final path.
```

    Hope this helps

    Reply

34. Max Says:

    2010-06-03 at 20:57
    Hi lalith,

    Thanks for your reply,

    I have tested with your pseudocode and it helped me understand, thank you.
    I also found good pseudocode on wikipedia for doing that :
    http://en.wikipedia.org/wiki/Floyd%E2%80%93Warshall_algorithm
    Under “Path reconstruction”

    My only problem now is printing the path in the good order, sometimes it’s in the good order, sometimes not, I guess it has to do with the function being recursive, I’ll check to fix that now.

    Here is what my algorithm look so far:

```
//————————————————————————————
//print the shortest path between i and j
int getPath( int chemin[][size], int next[][size], int i, int j)
{
if (chemin[i][j] == 0 )
{
```

```
printf("\n no path \n");
return 0;
}

int intermediate = next[i][j];
if (intermediate == 0)
{
printf("TEST2 %d, ", intermediate);
printf(" ");
return 0;
}
else
{
printf("TEST3 %d, ", intermediate);
//printf("Mot : %s, ", tWords[intermediate]);
return getPath(chemin, next, i, intermediate) + intermediate + getPath(chemin, next,
intermediate, j);
}

}
```

Reply

35. Julio Says:

    2010-07-14 at 3:04
    Excellent! I was searching for a very detailed explanation! Super! Kudos for you

    Reply

36. ruhallah Says:

    2010-07-30 at 18:16
    thanks.

    Reply

37. sushant Says:

    2010-08-05 at 14:47
    thanks…

    Reply

38. shraddha Says:

    2010-11-10 at 2:42
    can this program be used to detect the negative cycle in the graph???
    i.e. can the diagonal elements of the output matrix be negative???but the for loop we used in program has i not equal to j hence i feel that diagonal elemnts of the output always remain 0 even if a negative cycle exists..please correct me if i am wrong

Reply

39. Gayatri Says:

    2010-11-28 at 20:02

    Thanks a ton :)

    Reply

40. imran Says:

    2010-11-30 at 9:28

    I would suggest you to do it in a better manner. Please don't confuse people with complicated logic. Please refer Analysis and design of Algorithms by Anany Levitin you will have a better idea of what exactly the concept is, hope you take it seriously.

    With regards,
    Md Imran Pasha

    Reply

41. mokhtar Says:

    2010-12-08 at 2:30

    please every body can help me
    how can i find shortest path if I have map inside it cities and i want to beginning from city to city I closest it and print the path from start to end

    Reply

42. Patrick Bezzina Says:

    2011-01-07 at 14:41

    Hi,
    I am Patrick and I am going to do my thesis about shortesting path finding and I am going to use the Floyd-Warshall Algorithm and I found this informations very useful.
    My problem is, what do you mean by interspace A between any two nodes.

    Thanks very much,
    Patrick.

    Reply

    1. Aravind Rao Says:

       2011-02-13 at 9:02

       It means that if you want to go from Node C to Node D, see if Node C to Node A + Node A to Node D is shorter than Node C to Node D. Basically, you are 'inter spacing' Node A between Node C and Node D.

       Reply

43. Hamed Says:

2011-04-20 at 0:06
This was extremely helpul, thanks

Could you bring more complicated examples so that we see how the algorithm works in more general situations?….

Regards,

Reply

14. abhi shek das Says:

2011-04-26 at 11:43
for 5 nodes…you may increse the nodes no..so as accordingly the values in for loops…the simplest one

```
#include
#include
void main()
{
int d[6][6],n,i,j,k;
printf("Enter the number of nodes:\n");
scanf("%d",&n);
printf("Enter the distance matrix:\n");
for(i=0;i<5;i++)
{
for(j=0;j<5;j++)
{
scanf("%d",&d[i][j]);
}
}
for(k=0;k<n;k++)
{
for(i=0;i<n;i++)
{
for(j=0;j=(d[i][k]+d[k][j]))
d[i][j]=d[i][k]+d[k][j];
}
}
printf("The d%d:\n",k);
for(i=0;i<5;i++)
{ for(j=0;j<5;j++)
{ printf(" %d ",d[i][j]);
}
printf("\n");
}
}

printf("The required shortest path is d%d:\n",n);
for(i=0;i<5;i++)
{ for(j=0;j<5;j++)
{ printf(" %d ",d[i][j]);
}
printf("\n");
}
getch();
}
```

Reply

15. haritha Says:

    2011-05-02 at 15:47
    Thanks for the great help.

    Reply

16. Somesh Says:

    2011-05-27 at 20:04
    Hey..thnx tons…jus wt i ws lukn fr..an informal discussions ascompared to all other lame FORMAL explanations available..jus 1 thing…the matrix column indexing is a bit shiftes which resulted in some confusion

    Reply

17. Somesh Says:

    2011-05-27 at 20:06
    Hey..thnx tons…jus wt i ws lukn fr..an informal discussion as compared to all other lame FORMAL explanations available..jus 1 thing…the matrix column indexing is a bit shifted which resulted in some confusion…

    Reply

18. Nathan Yeung Says:

    2011-11-01 at 23:58
    Does the interspacing including the previous vertices? Such that after the interspace of A, for interspace of B also includes the interspace of A?

    Thanks,
    Nathan

    Reply

49. indierobot Says:

    2011-11-18 at 2:28
    Thank you for this! Your pseudocode presentation is much clearer than the presentation in Cormen et al.'s Algorithms. Helped me through a homework problem :-) Thanks again!

    Reply
50. sukanta dasgupta Says:

    2011-11-26 at 15:52
    Very useful algo. it helps me a lot…
    thank you very much.

    Reply
51. Anurag Says:

    2012-04-02 at 14:51
    Thanks a lot!! u helped me to complete my term project! :-)

    Reply
52. meledy Says:

    2012-09-15 at 10:04
    this is a bit complicated….plz make it eay

    Reply
53. Jasmine Says:

    2012-09-20 at 19:16
    What is the difference between single source shortest path(Dijkstra's alg) and all pairs shortest path(Warshall)?

    Reply
54. S.k. Pandey Says:

    2012-10-12 at 17:04
    NICE EXP.

    Reply
55. Kashish Says:

    2012-11-19 at 20:30
    just the right place to revise before your test! :)

    Reply
56. ankita Says:

2013-05-11 at 11:15
how to intespace any node

Reply

57. vishal Says:

    2013-05-16 at 11:50
    thanx a lot…..

    Reply

58. Tharindu_DG Says:

    2013-09-05 at 8:15
    There is a negative cycle (B->E->D) of cost -5. There for does this example valid? You could go from A to C with a cost of minus infinity. i.e first go from A to B then go through the B->E->D->B cycle infinite times and then go to C.

    Reply

59. Jerilyn Says:

    2014-10-09 at 9:51
    I read a lot of interesting articles here. Probably you spend a lot
    of time writing, i know how to save you a lot of time,
    there is an online tool that creates readable, SEO friendly
    posts in minutes, just search in google – masagaltas free content

    Reply