Forums / Programming Assignments / Programming Assignment #4 Help Center

PA#4: some small test cases

Subscribe for email updates. PINNED

No tags yet. + Add Tag

Sort replies by: Oldest first Newest first Most popular

Tim Roughgarden INSTRUCTOR · a month ago PINNED

Below are some small test cases. I encourage learners to develop and post larger ones, as needed.

By Jason Semko

```
1 4
2 8
3 6
4 7
5 2
6 9
7 1
8 5
8 6
9 7
9 3
```

```
Answer: 3,3,3,0,0
```

===

```
1 2
2 6
2 3
2 4
3 1
3 4
4 5
5 4
6 5
6 7
7 6
7 8
8 5
8 7
```

```
Answer: 3,3,2,0,0
```

===

```
1 2
2 3
3 1
3 4
5 4
6 4
8 6
6 7
7 8
```

```
Answer: 3,3,1,1,0
```

==

```
1 2
2 3
3 1
3 4
5 4
6 4
8 6
6 7
7 8
4 3
4 6
```

```
Answer: 7,1,0,0,0
```

==

```
1 2
2 3
2 4
2 5
3 6
4 5
4 7
5 2
5 6
5 7
6 3
6 8
7 8
```

```
7 10
8 7
9 7
10 9
10 11
11 12
12 10
```

```
6,3,2,1,0
```

⬆ 20 ⬇ · flag

Pavel Velikhov · a month ago 🔗

Got all the small test cases right, but get a wrong answer on the real problem... There must be some tricky thing to the real problem that's not in these small tests.

⬆ 3 ⬇ · flag

mugnaio · a month ago 🔗

notice you have to get the 5 higher results in real problem if they are more than 5

⬆ 0 ⬇ · flag

Pavel Velikhov · a month ago 🔗

Yup, that's what I do

⬆ 0 ⬇ · flag

A Fern · a month ago 🔗

Hmm, I'm also getting all the small test cases right, but my final solution to the assignment input is failing :(

⬆ 0 ⬇ · flag

A Fern · a month ago 🔗

Ok, I solved my problem, I had a bug to do with when I was setting a node to be explored - i.e. I was setting it as explored only after I popped it from the stack, whereas I should have been setting it to explored before pushing it onto the stack!

⬆ 4 ⬇ · flag

Pavel Velikhov · a month ago 🔗

Doh! Had the same exact problem. For some reason it seemed much more reasonable to mark the node as visited after popping it off the stack.... Now it works correctly

↑ 0 ↓ · flag

A Fern · a month ago

@Pavel - glad you fixed yours too! It's definitely a subtlety of the algorithm!

↑ 0 ↓ · flag

Pavel Velikhov · a month ago

Hmm... I think it still makes sense to mark the node as visited after popping it off the stack (because that's where we're doing something with that node).
However, we had a bug in the sense that we were pushing the same node into the stack multiple times. So maybe we should have just guarded the stack against duplicates.

↑ 0 ↓ · flag

Pavel Velikhov · a month ago

Hmm... I think it still makes sense to mark the node as visited after popping it off the stack (because that's where we're doing something with that node).
However, we had a bug in the sense that we were pushing the same node into the stack multiple times. So maybe we should have just guarded the stack against duplicates.

↑ 0 ↓ · flag

Takafumi Yoshida · a month ago

Hi, Pavel!
*Yeah, a **stackedFlag** did my heart good!*
Best,
Taka

↑ 1 ↓ · flag

alan kilmartin · 25 days ago

```
1 4
2 8
3 6
4 7
5 2
6 9
6 5
7 1
8 5
8 6
9 7
9 3

This is the first example and i have added "6 5" to it.  Is the answer
```

```
6,3,0,0,0 or 6,3,3,3,0 or neither maybe
```

↑ 1 ↓ · flag

Philip Ronan · 25 days ago

I got "6,3,0,0,0".

6,3,3,3,0 is very unlikely to be correct, given that there are only 9 nodes in the graph.

↑ 0 ↓ · flag

alan kilmartin · 25 days ago

got it now, wasn't sure if you could get an SC within an SC. i guess i should of finished watching the lecture before i asked. =).

thanks for the confirmation

↑ 1 ↓ · flag

Nikhil Satish · 25 days ago

I hit upon the same 'explored' flag problem. I think instead that the meaning of the flag can be modified- to more accurately describe what it's doing- to 'schedule to be explored' since that is what we're doing when we push to the stack. This way it makes sense and we don't have to add code to force the flag to make sense in a round about manner.

↑ 0 ↓ · flag

Ahmed Yossre · 23 days ago

@Alan Kilmartin

1 2
2 4
3 1
3 4
4 5
5 3

Do the previous output
5 3 0 0 0
or
5 0 0 0 0
??

↑ 1 ↓ · flag

Adam Balogh · 22 days ago

I also ran into the problem with setting a node to seen after taking it off from the stack, not when pushing it to the stack (should have read the forum...).
Here is a test case:
1 2
1 3
2 1
2 4
3 2
4 1
If you get 5,0,0,0,0 instead of 4,0,0,0,0 then this could be the reason.

⬆ 2 ⬇ · flag

+ Comment

Philip Ronan · a month ago 🔗

I wrote a Python script that can generate test data with up to about 10,000 nodes. I hope somebody finds it useful :-)

```
from __future__ import print_function
import random
max_node = 1
sccs = [[1]]
edges = []

def add_scc():
  global max_node, sccs, edges
  max_node += 1
  for scc in random.sample(sccs,random.randint(1,min(3,len(sccs)))):
    edges += [[random.choice(scc),max_node]]
  sccs += [[max_node]]

def expand_scc():
  global max_node, sccs, edges
  max_node += 1
  which_scc = random.randint(0,len(sccs)-1)
  if len(sccs[which_scc])==1:
    old_node = sccs[which_scc][0]
    sccs[which_scc] += [max_node]
    edges += [[old_node,max_node],[max_node,old_node]]
  else:
    (node1,node2) = random.sample(sccs[which_scc],2)
    if edges.count([node1,node2]):
      edges.remove([node1,node2])
      edges.append([node1,max_node])
      edges.append([max_node,node2])
      sccs[which_scc].append(max_node)
    elif edges.count([node2,node1]):
      edges.remove([node2,node1])
      edges.append([node2,max_node])
      edges.append([max_node,node1])
      sccs[which_scc].append(max_node)
    else:
      edges.append([node1,max_node])
      edges.append([max_node,node2])
      sccs[which_scc].append(max_node)

def build_graph():
  global max_node, sccs, edges
  while True:
    nscc = int(raw_input('How many SCCs? '))
    if nscc>=1:
      break
    print('Must have at least one. Try again.')
  while True:
    add_nodes = int(raw_input('How many additional nodes? '))
    if add_nodes>=0:
      break
    print('No. Try again.')
  for i in range(nscc-1):
    add_scc()
  for i in range(add_nodes):
    expand_scc()
  scc_sizes = [len(s) for s in sccs]
  scc_sizes.sort()
  scc_sizes.reverse()
  random.shuffle(edges)
  print('Graph created. SCC sizes are as follows:')
  print(','.join([str(s) for s in scc_sizes]))
  outfile = raw_input('Output filename: ')
  with open(outfile,'w') as the_file:
    for e in edges:
      the_file.write(str(e[0])+' '+str(e[1])+'\n')
  print('Finished')

build_graph()
```

When the script runs, it will ask you how many SCCs you want to create, and how many additional nodes you want added to the graph. The additional nodes are added to randomly selected SCCs. For example:

```
$ python create_test_data.py
How many SCCs? 3
How many additional nodes? 7
Graph created. SCC sizes are as follows:
5,3,2
Output filename: test_data.txt
Finished
```

Contents of output file:

```
1 2
5 6
1 4
7 8
4 10
3 5
2 9
6 3
10 8
2 3
9 2
8 1
4 7
```

↑ 19 ↓ · flag

Jon Hoffman · a month ago

That's great, I was going to do the same in c++ just to play around but I have not had time. Is this compatible with Python 3.4?

↑ 0 ↓ · flag

Philip Ronan · a month ago

Hi Jon, I just tried it out at ideone.com, and it looks like it will run in Python 3 if you replace every occurrence of `raw_input` with `input`.

↑ 1 ↓ · flag

Nilesh · a month ago

Thanks.
But i still can't figure out the bug in my program, all the test cases given here work and all the test cases generated by your script works too. IDK what should i do...

EDIT: figured out the problem. Notepad++ was not properly converting UNIX file-endings to windows file-endings. So i was getting wrong input.

↑ 0 ↓ · flag

Philip Ronan · a month ago

Hi Nilesh,

Performing a depth-first search on the actual PA#4 data involves *a lot* of recursion, so maybe you're getting incorrect results due to a stack overflow or a buffer overrun. Are you getting the same results every time? If not, then perhaps you're getting undefined behaviour for some reason.

Here's an idea: try starting the initial DFS from a node selected at random. That way you might be able to traverse the graph without so much recursion.

↑ 0 ↓ · flag

Jon Hoffman · a month ago 🔗

@Philip
This is the first time I have run python in a few months. Your code works well in python 3.4 with that minor change. First I created just a few thousand nodes and it worked and ran fine. Then I told it to create 123,456,789 nodes. It is still running...

↑ 0 ↓ · flag

Philip Ronan · a month ago 🔗

Haha. Yes, that will take a while. The script isn't super-efficient and will start to choke if you ask it for a graph with more than about 10k nodes. I'll go back and edit my earlier post :-D

↑ 0 ↓ · flag

Jon Hoffman · a month ago 🔗

@Philip
Yeah, I pushed it too far. I stopped it after a few hours of it running in the background. I ran a 100k node test and it was not too bad. It might be easy to modify the code slightly such that several 100k runs could be stitched together to create a very large file, but not for me - I don't python.

↑ 0 ↓ · flag

Guangliang He · a month ago 🔗

I coded the recursive DFS in java. I'm getting all test cases correct, on the SCC.txt, I'm getting stack overflow error. I tried to increase jvm stack space but only gets more recursions, still get stack overflow. :(

↑ 0 ↓ · flag

Samilenko Vadym · a month ago 🔗

Hi, Philip!

Your generator relly help me! My code worked perfect with test cases from Tim Roughgarden but complex SCC.txt failed!
For me this example helped to find a bug:

```
1 9
2 3
1 12
10 5
4 1
```

```
13 4
2 7
1 2
9 10
11 8
5 4
8 3
12 10
1 3
7 6
9 13
6 2
3 11
```

Correct answer:

```
7,3,3,0,0
```

After that I tested about 5 different test cases generated by your tool and they were computed correctly (and SCC.txt finaly). Many thanks for saved time!

↑ 5 ↓ · flag

Madan Kapoor · a month ago 🔗

Thanks Dude ,
I used a 7100 nodes graph generated from your test case and it worked well.
I got one attempt wrong because i entered the answer wrongly in the coursera answer page , your test case helped me to find my mistake .

↑ 0 ↓ · flag

Gang Peng · a month ago 🔗

This test case helped me to find the bug in my code. Turns out that I visit the nodes in exactly the reverse order on second pass since I recorded the completed nodes in a vector and then visit the nodes in the vector one by one in the outer looper. Suprisingly enough my code worked on all small tests cases posted by Tim. :)

↑ 0 ↓ · flag

Val Miscenko · a month ago 🔗

Philip, thanks a lot for test generating tool (very helpful).

But my problem is - all tests except SCC.txt are passing just fine. Kinda lost :(
Already tried unix2dos - the same thing - don't see a problem here. Using python.

EDIT:
:) Finally found - I was incorrectly creating second graph - it worked only for sequential vertices numbers and surprisingly for all generated tests. But it failed for simple cases like:
1 2
3 1

and

1 2
 5 10

↑ 0 ↓ · flag

+ Comment

Joseph Kim · a month ago

Seems like a lot of people are saying they get the small tests correct, but not the actual SCC.txt problem.
I thought that was my case as well, until I realized that my iterative DFS implementation would give me wrong answers if I start it at certain nodes, even for the small tests.

My python implementation uses a dictionary of tail nodes mapping to lists (apparently the suggested way to implement graphs in python by Guido van Rossum) containing the heads of the directed edges, so the start node of my DFS algorithm is kind of random.

So when I run something like this (scc.py reads in the first argument 'text.txt' that contains the graph edges):

```
for i in {1..1000}; do python scc.py test.txt ; done | sort -u | wc
```

Sometimes, I would get more than one solution to the same problem, which means that there is something wrong with the algorithm that determines the **order** in which the strongly connected components are detected. =)

So maybe for some of you guys, if you haven't tried it, try running it multiple times to see if you're always getting the same answer even for the small test cases.

Two other things that I did, which was very helpful:

1) Write out to a log file that records which nodes you see for the first time in the DFS

2) If you're using python, use the python package pydot, or pydotplus for python3 to visualize the graphs and the reverse graphs.

↑ 1 ↓ · flag

+ Comment

Jacob Taber · a month ago

This test clued me in that my code was not handling the situation where there were two vertices, outside of any multi-vertex SCC, with just an edge between them.

```
1 2
2 3
3 1
3 4
5 4
6 4
8 6
6 7
7 8
```

```
Answer: 3,3,1,1,0
```

Thank you for the tests Jason Semko

↑ 9 ↓ · flag

cheng-chih yang · a month ago 🔗

Thanks! My program gives me 3, 3, 2, 0, 0 for this test set, and by hand I can see that it should be 3, 3, 1, 1, 0. But I still cannot figure out why....any pointers?

↑ 0 ↓ · flag

cheng-chih yang · a month ago 🔗

Never mind...I figure out the reason. It's because I pop the parent before some of the children are still in the stack. Thanks a lot guys!

↑ 0 ↓ · flag

Phillip Henry · 24 days ago 🔗

Thanks for this. Like a lot of people, my code was passing all the other tests but failing on the actual task. This highlighted the problem in my code (that is, using a naive stack implementation).

↑ 0 ↓ · flag

Ahmed Yossre · 24 days ago 🔗

LOL I got 3 1 1 1 1

↑ 0 ↓ · flag

Ahmed Yossre · 23 days ago 🔗

Thanks GOD .. got it like you now :D

↑ 0 ↓ · flag

🗑A post was deleted

+ Comment

Paul Nordin · a month ago

Just a conceptual question. Is the second test case actually a valid directed graph? There are a few arcs with tail <==> head. For example, is 4 <==> 5 a valid path?

```
4 5
5 4
```

↑ 0 ↓ · flag

Guangliang He · a month ago

That should be OK. In the SCC.txt, there is even a self loop 1 1 (line 1).

↑ 0 ↓ · flag

Jon Hoffman · a month ago

@Guangliang
I don't even load lines that "self loop" and the code works, fwiw.

↑ 0 ↓ · flag

+ Comment

Ananth · a month ago

All the small test cases work fine and I get the correct answer but the final answer is wrong. Not sure what I'm missing. I'm writing this in Java. Does anyone have any other suggestions to check ? I'm setting the node to visited before pushing in the stack. And I have gone over the code multiple times and it seems to be correct.

↑ 0 ↓ · flag

Samilenko Vadym · a month ago

Hi Ananth,

Generate different test cases using tool on Python writed by Philip. If SCC,txt computed wrong than you will definetly generate small graph that computed also incorect. When you find in just draw it on a list and solve problem step by step by yourself. After that debug and you will find a bug.

Have the same problem and it worked for me!
Good luck!

↑ 0 ↓ · flag

Ananth · a month ago

Thanks Samilenko. I will try that today and I hope I will be able to find the bug.

↑ 0 ↓ · flag

Ananth · a month ago

hi Samilenko,

I was able to find the bug. thanks so much.

↑ 0 ↓ · flag

+ Comment

Ananth · a month ago

Thanks Philip Your tool to generate graphs helped me to find a bug.

This is the graph that helped me identify my bug

```
3 4

1 3

7 4

2 3

4 5

4 6

1 2

2 4

1 4

6 7
```

Correct answer

```
3,1,1,1,1
```

My code had a bug where it was backtracking.

↑ 2 ↓ · flag

A post was deleted

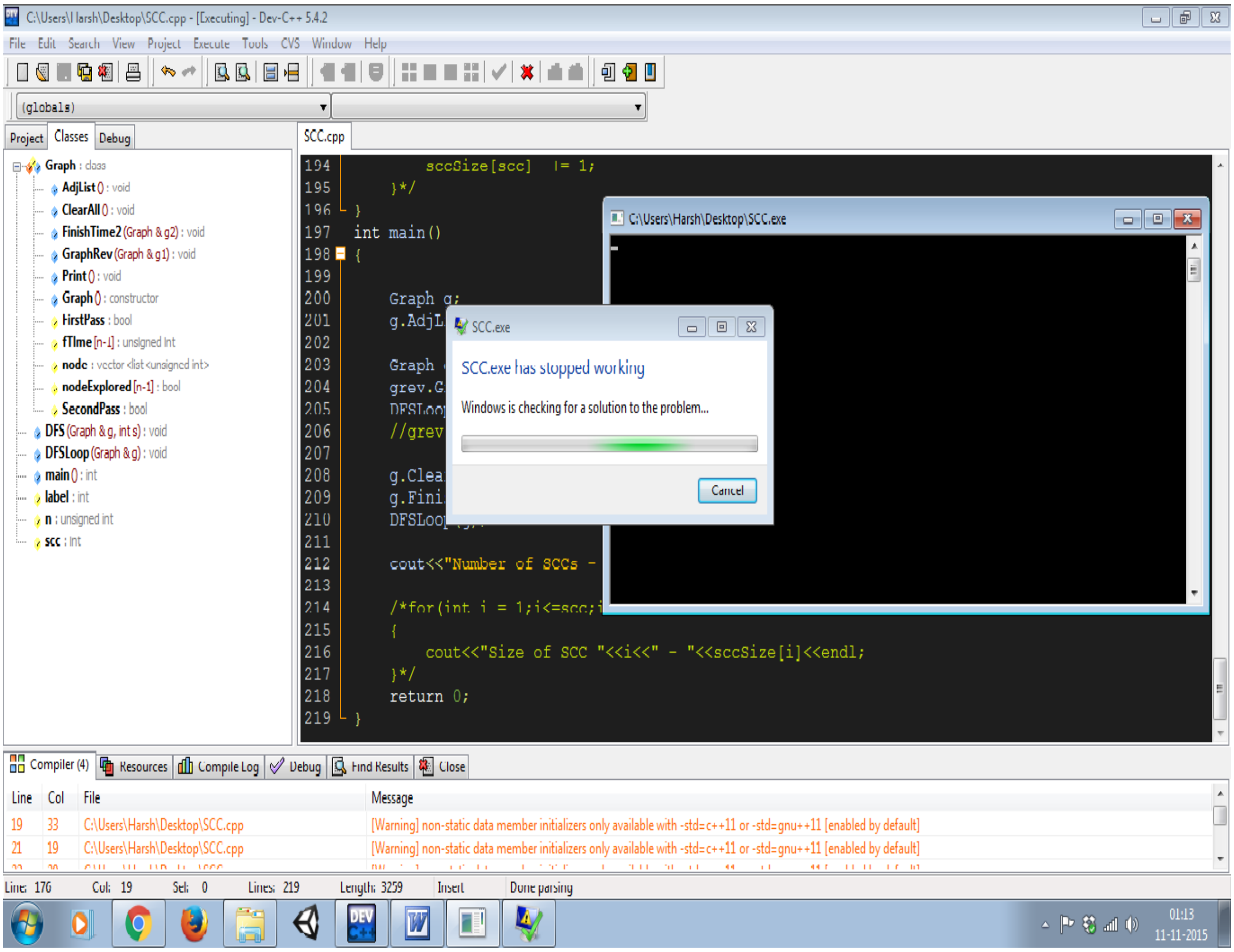
Melanie Day · 24 days ago

This found a bug for me where I was counting nodes that were pointed to by other nodes but pointed to nothing multiple times for every node that pointed to them...

0 · flag

+ Comment

Harsh Chaudhari · a month ago


C:\Users\Harsh\Desktop\SCC.cpp - [Executing] - Dev-C++ 5.4.2
File Edit Search View Project Execute Tools CVS Window Help
(globals)
Project Classes Debug
SCC.cpp
Graph : class
AdjList () : void
ClearAll () : void
FinishTime2 (Graph & g2) : void
GraphRev (Graph & g1) : void
Print () : void
Graph () : constructor
FirstPass : bool
fTime [n-1] : unsigned int
node : vector <list <unsigned int>
nodeExplored [n-1] : bool
SecondPass : bool
DFS (Graph & g, int s) : void
DFSLoop (Graph & g) : void
main () : int
label : int
n : unsigned int
scc : int
C:\Users\Harsh\Desktop\SCC.exe
SCC.exe
Cancel
Compiler (4) Resources Compile Log Debug Find Results Close
Line: 176 Col: 19 Sel: 0 Lines: 219 Length: 3259 Insert Done parsing
01:13
11-11-2015

```
            sccSize[scc]  != 1;
        }*/
}
int main()
{

    Graph g;
    g.AdjL

    Graph
    grev.G
    DFSLoo
    //grev

    g.Clea
    g.Fini
    DFSLoo

    cout<<"Number of SCCs -

    /*for(int i = 1;i<=scc;i
    {
        cout<<"Size of SCC "<<i<<" - "<<sccSize[i]<<endl;
    }*/
    return 0;
}
```

SCC.exe has stopped working

Windows is checking for a solution to the problem...

Line	Col	File	Message
19	33	C:\Users\Harsh\Desktop\SCC.cpp	[Warning] non-static data member initializers only available with -std=c++11 or -std=gnu++11 [enabled by default]
21	19	C:\Users\Harsh\Desktop\SCC.cpp	[Warning] non-static data member initializers only available with -std=c++11 or -std=gnu++11 [enabled by default]

My Code Works For Smaller Test Cases But It Just crashes when I Run the Original File. Could Memory Limitation be the issue in my case ?

0 · flag

+ Comment

Ananth · a month ago

Hi Harsh,

Are you seeing any error messages or stack trace when the code crashes? If you look closely at those logs you should be able to identify why it crashed.

Thanks,
Ananth

↑ 0 ↓ · flag

+ Comment

Harsh Chaudhari · a month ago 🔗

It shows an error msg SIGSEGV 'Segmentation Fault' When I Debug It.

↑ 0 ↓ · flag

Ananth · a month ago 🔗

Hi,

I'm not familiar with C++
but I saw this post on stackoverflow which points to memory corruption
http://stackoverflow.com/questions/5856628/program-received-signal-sigsegv-segmentation-fault

↑ 0 ↓ · flag

Nilesh · a month ago 🔗

add this to linker options:
"-Wl,--stack,268435456"
It will set the stack limit to 256 MB which would be enough

↑ 1 ↓ · flag

Anton Malmygin · 6 days ago 🔗

Thanks, Nilesh! Really helped me.

Btw, in my case it was "-Wl,-stack_size,2000000" as I used C++ and Xcode 7

↑ 0 ↓ · flag

+ Comment

Harsh Chaudhari · a month ago 🔗

Ananth ,did you face any problem regarding memory limitation when storing the graph on stack memory (for vertices 1-875714).?

↑ 0 ↓ · flag

Ananth · a month ago 🔗

Harsh, I had stackOverFlow problem when I implemented the algorithm using recursion, but I didn't have this problem when I used iterative approach. I'm using Java.

🡅 0 🡇 · flag

+ Comment

Cavelle Benjamin · a month ago

Ran the algorithms with DFS-Recursion and ran into segmentation fault. Switched to DFS-Loop. Runs to completion. The additional examples helped find errors

🡅 0 🡇 · flag

+ Comment

Arpit Gagneja · a month ago

I am using Stack peek method when assigning explored status to the vertex. After that, I am using push method for adjacent vertex. I am actually using pop when the vertex is about to finish i.e. while having its finishing time set. Is it right way to do?

🡅 0 🡇 · flag

+ Comment

Harsh Chaudhari · 25 days ago

I found the error but I do not know what can I change to make it work. I am not able to store 875714 vertices even though I am storing it on the Heap. Any Suggestions as to what might be wrong ? (PS: It is working on all of the above smaller test cases).

```
class Graph
{
        public:
        list<int> *node;                                        // n =875715(nodes+1)
        bool nodeExplored[n-1] = {false};
        unsigned int fTime[n-1];
        bool FirstPass = true;
        bool SecondPass = false;

        Graph();
        void AdjList();
        void GraphRev(Graph & g1);
        void ClearAll();
        void FinishTime2(Graph & g2);
        void Print();


};
```

```
Graph :: Graph()
{
        node = new list[n];                     //n=875715 (nodes+1).
}

void Graph :: AdjList()
{
        int a,b;
        ifstream file;
        file.open("SCC1.txt");                       //SCC1.txt (is the input file of 875715 vertices).
        string line;
        while (getline(file, line))
                {
                        istringstream iss(line);
                        while ((iss >> a>> b))
                        {
                                node[a].push_back(b);
                        }
                }
        file.close();
}
```

↑ 0 ↓ · flag

+ Comment

Venkatesh Arjunan · 25 days ago

1. I used the first test case here and after that worked on SCC.txt. Actually drawing on a piece of paper, the graph, the reverse graph, finishing times for the test case and then used it for debugging my code
2. i created the reverse graph while reading from the file itself
3. i used map with node as key and linkedlist as value so that transforming the node value is easier
4. iterative DFS (key point - Pop the parent last)

HTH and enjoy !

↑ 0 ↓ · flag

Venkatesh Arjunan · 25 days ago

Although i calculated the "leaders" as mentioned in the lecture, i am not sure we need to use it .. For example , for the first test case here with 9 vertices , i got only two leaders 9 and 7 but the SCC's are three. So the statement "SCCs have same leader as value" is wrong ?

So in my code, in both the outer loops, i started from last labelled vertex to 1 in a FOR loop and it worked well (i mean i did not use leaders for the 2nd DFS)

↑ 0 ↓ · flag

+ Comment

Sharif Khan · 25 days ago 🔗

my c++ program stopped working.when input the large SCC.txt file.
but it working properly at all the test cases above.
someone kindly help me

↑ 0 ↓ · flag

Ahmed Yossre · 24 days ago 🔗

what is the problem exactly when you debug it ?

↑ 1 ↓ · flag

Sharif Khan · 24 days ago 🔗

its cant handle the huge input.it cant store the scc.txt file.

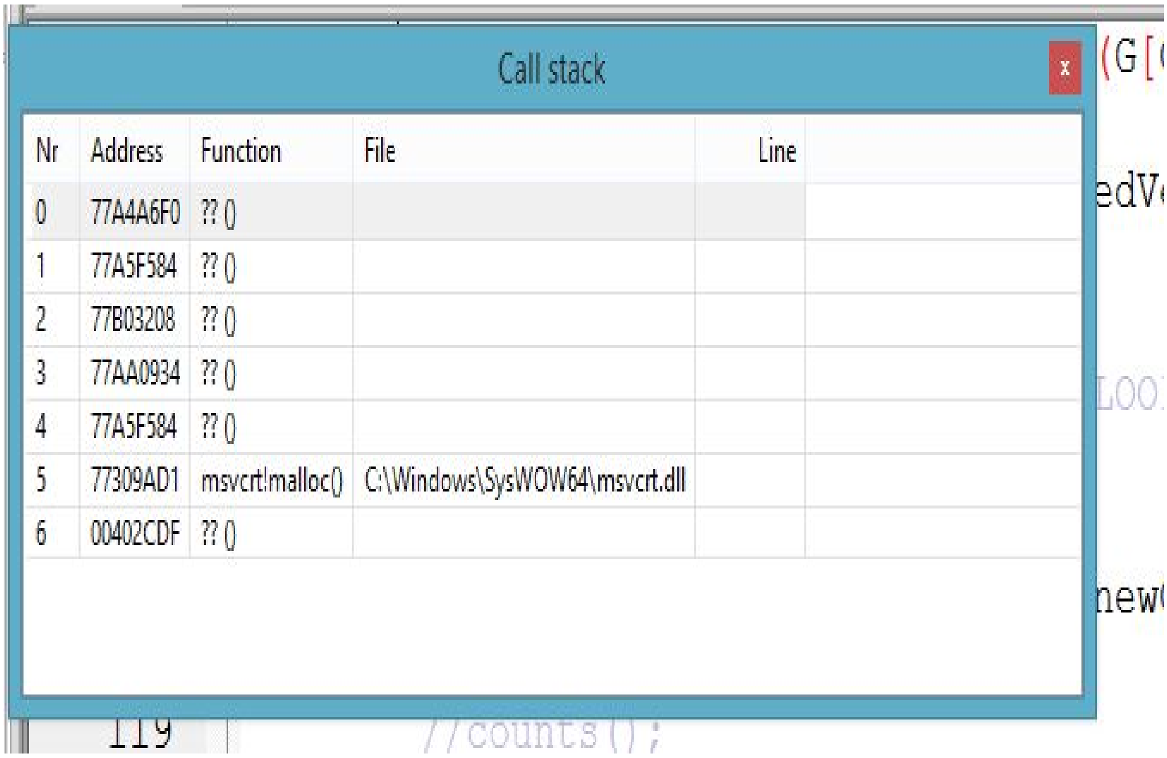
↑ 0 ↓ · flag

Ahmed Yossre · 23 days ago 🔗

Yes, that's the result, what is the error ... debug it .. see what's the error exactly
One of the possible errors that you're using Recursive functions which isA will yield a stackoverflow exception

↑ 0 ↓ · flag

Sharif Khan · 23 days ago 🔗



Nr	Address	Function	File	Line
0	77A4A6F0	?? ()		
1	77A5F584	?? ()		
2	77B03208	?? ()		
3	77AA0934	?? ()		
4	77A5F584	?? ()		
5	77309AD1	msvcrt!malloc()	C:\Windows\SysWOW64\msvcrt.dll	
6	00402CDF	?? ()		

when i debug my programm.i get this error.i think its memory allocation problem.

0 · flag

+ Comment

Cindy Van Dooren · 24 days ago

I spent the morning debugging my code by using all your test cases. My code now passes all of them. Still, I didn't pass the assignment. Since I'm all out of opportunities to try (I've tried 5 times, because I thought I had the answer every single time), I was wondering if someone could share their solution with me (in private, or do we receive the answers at some point?), so that I know what the goal is when trying to debug my code again. Tips are welcome as well.

0 · flag

Ahmed Yossre · 23 days ago

try this
1 2
2 1
3 4
4 3
4 1

op: 4,0,0,0

&

1 2
2 4
4 5
5 3

3 4
3 1
this should output 5,0,0,0,0

↑ 0 ↓ · flag

Marek Beseda · 22 days ago

The first one's answer should be 2,2,0,0,0 as far as i know

↑ 1 ↓ · flag

+ Comment

Martin Schultze · 24 days ago

My code works on all example instances and returns the correct answers. Sadly it results in stack overflow in R (the language I'm using). I've tried several workarounds, e.g. by increasing the pointer protection stack to the allowed maximum. It then takes like 30 minutes before overflowing again. Has anyone actually been able to complete the assignment in R?

↑ 0 ↓ · flag

Adam Balogh · 22 days ago

I have used Matlab which I think should be quite similar. I have also hit a recursion limit and had to convert the code to iterative DFS.
I don't know about R, but Matlab does not have a native stack capability, so that had to be implemented as well using an array and a counter (to avoid dynamically resizing the array).

↑ 0 ↓ · flag

+ Comment

Ahmed Yossre · 24 days ago

In the example:

1 2
2 6
2 3
2 4
3 1
3 4
4 5
5 4
6 5
6 7
7 6
7 8

8 5
8 7

why the answer is not:
3,3,2,1,1 ??

the nodes 4 & 5 are left alone ?

↑ 0 ↓ · flag

Ahmed Yossre · 23 days ago

ok, Thanks GOD I now have it 3,3,2,0,0
But I still cannot understand why it's not 3,3,2,1,1 ..

↑ 0 ↓ · flag

Matteo Nicoli · 23 days ago

because you have only 8 vertices: a vertex can only be assigned to one SCC.

↑ **1** ↓ · flag

+ Comment

⬇ scroll down for more ⬇