PA#5 Test Cases

No tags yet. + Add Tag

Sort replies by: Oldest first Newest first Most popular

Tim Roughgarden INSTRUCTOR · a month ago

By TP Diffenbach:

Note that the output given is not the same as the answer to the homework question, instead it is "node, cost to arrive at node, path to node from node 1"

input:
1 2,3 3,3
2 3,1 4,2
3 4,50
4

output:
1 0 []
2 3 [2]
3 3 [3]
4 5 [2, 4]

---
input:
1 2,3 3,5
2 3,1 4,2
3 4,50
4

output:
1 0 []
2 3 [2]
3 4 [2, 3]
4 5 [2, 4]

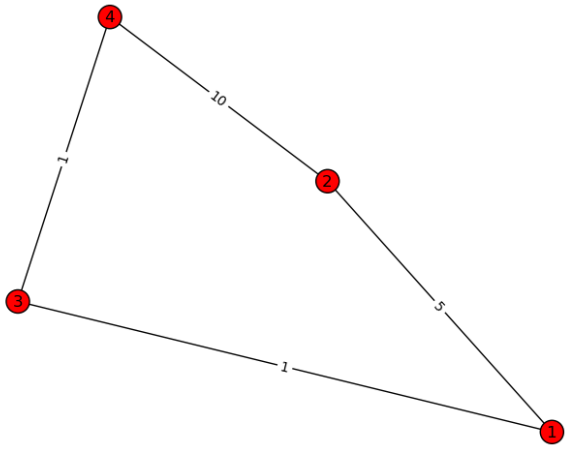
---
input:
1 2,3 3,4
2 3,1 4,2
3 4,50
4

output: underspecified, as there is a tie. Presumably, you'd choose the path with fewest nodes, but who cares?

By Drew Verlee

****

Test Case small:



input:

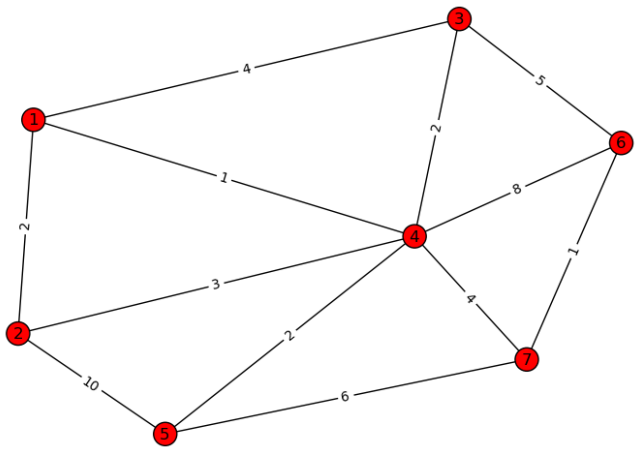
1 2,5 3,1
2 4,10 1,5
3 4,1 1,1
4 2,10 3,1

start = 1
finish = 4

expected cost: 2

****

Test Case medium:



input:

1 2,2 4,1
2 4,3 5,10
3 1,4 6,5
4 3,2 5,2 6,8 7,4
5 7,6
7 6,1

start = 1
finish = 7

expected cost: 5

****

Test Case Large:

Picture: not very helpful i'm afraid...

input:

0 1,28 2,4 3,53 4,95 5,59 7,48 8,84 10,27 11,7 12,91 13,96
1 0,28 2,79 5,30 7,59 8,62 9,20 11,15 12,16 13,96
2 0,4 1,79 3,80 5,19 6,74 8,59 10,80 11,81 12,65 13,7 14,13
3 0,53 2,80 4,87 5,42 6,70 7,8 12,44 13,92
4 0,95 3,87 5,86 6,46 7,44 9,12 10,39 12,14 13,95 14,19
5 0,59 1,30 2,19 3,42 4,86 6,12 7,0 8,72 10,92 11,54 12,0 13,68 14,22
6 2,74 3,70 4,46 5,12 7,86 9,76 10,68 11,32 12,62 13,34
7 0,48 1,59 3,8 4,44 5,0 6,86 8,38 9,34 10,12 11,55 12,25 14,9
8 0,84 1,62 2,59 5,72 7,38 10,66 11,31 12,95
9 1,20 4,12 6,76 7,34 10,63 11,37 12,93 13,79
10 0,27 2,80 4,39 5,92 6,68 7,12 8,66 9,63 12,45 14,78
11 0,7 1,15 2,81 5,54 6,32 7,55 8,31 9,37 12,75 13,87
12 0,91 1,16 2,65 3,44 4,14 5,0 6,62 7,25 8,95 9,93 10,45 11,75 13,46 14,63
13 0,96 1,96 2,7 3,92 4,95 5,68 6,34 9,79 11,87 12,46
14 2,13 4,19 5,22 7,9 10,78 12,63

start = 13
finish = 5

expected cost: 26

****

Largest case:

the cost is 9 with path [28, 16, 6] with start point 28 and finish 6. There is more in my comment below...

```
0 1,37 2,98 5,64 6,36 8,27 9,5 10,71 12,28 13,4 14,3 15,9 16,1 18,40 20,40 21,21 24,28
25,16 26,0 27,22 28,19 33,0 35,23 36,18 37,5 39,9 42,46 43,10 44,74 45,77 47,29 49,88
1 0,37 2,64 3,79 5,36 6,24 10,59 12,93 13,5 14,77 15,89 18,7 19,61 20,7 21,45 22,86 2
4,98 25,84 26,10 27,21 29,75 30,9 31,13 32,34 33,2 34,98 35,33 37,67 38,59 39,73 40,97
41,46 43,58 44,47 45,11 47,31 48,23 49,4
2 0,98 1,64 4,9 6,75 7,97 8,78 9,34 11,13 12,2 13,30 14,16 17,45 18,71 20,45 22,91 24,7
0 25,13 27,94 29,32 31,58 32,59 33,44 34,43 35,38 36,38 38,53 39,31 41,33 42,4 45,85 4
6,29 49,88
3 1,79 4,13 5,74 6,59 7,20 8,92 10,53 11,67 12,38 13,69 14,21 15,55 17,79 18,58 21,0 2
2,75 24,89 25,35 26,30 28,51 29,80 32,69 33,34 35,6 40,74 41,53 42,68 44,13 45,9 46,74
47,56 48,89 49,76
4 2,9 3,13 5,58 8,6 9,95 10,50 11,26 12,8 13,55 14,59 18,76 19,16 20,77 21,38 22,72 2
3,17 24,48 27,58 29,5 30,87 31,75 32,8 34,45 36,31 37,14 39,55 42,16 43,66 44,13 45,21
46,49
5 0,64 1,36 3,74 4,58 7,74 8,39 9,18 10,64 13,70 15,99 16,54 17,14 18,32 21,4 22,19 2
3,47 24,35 25,17 26,43 28,45 29,85 30,36 32,46 34,97 38,46 41,43 43,6 44,85 45,4 47,98
6 0,36 1,24 2,75 3,59 7,36 9,86 10,24 12,47 13,72 15,2 16,8 17,37 18,48 20,31 21,52 2
3,69 24,70 26,87 27,27 30,8 31,92 32,54 33,84 34,33 35,31 37,74 38,70 39,10 40,59 41,13
42,80 43,91 45,3 48,15 49,35
7 2,97 3,20 5,74 6,36 8,4 9,15 10,66 11,39 14,60 15,28 18,51 19,58 20,1 21,2 26,57 28,8
6 29,91 35,96 36,41 37,43 38,80 41,92 43,40 45,70 46,35 47,93 48,50 49,98
8 0,27 2,78 3,92 4,6 5,39 7,4 10,91 11,91 13,3 14,57 15,83 16,45 17,63 19,85 22,6 25,59
27,47 30,77 35,86 37,26 39,11 42,92 43,99 46,45 49,51
9 0,5 2,34 4,95 5,18 6,86 7,15 10,5 11,45 12,24 13,26 14,67 15,6 16,84 17,3 20,96 21,83
22,43 24,28 26,61 27,24 28,54 29,4 30,87 32,50 33,24 34,37 35,46 36,5 37,78 38,89 39,27
40,19 42,34 45,35 46,80 48,87
10 0,71 1,59 3,53 4,50 5,64 6,24 7,66 8,91 9,5 11,41 12,8 13,61 14,73 15,96 17,52 19,71
21,3 22,62 24,11 26,84 27,50 29,52 30,16 31,73 33,61 34,49 35,32 36,57 37,8 39,25 40,66
41,38 47,84 48,20 49,59
11 2,13 3,67 4,26 7,39 8,91 9,45 10,41 12,91 13,57 15,84 16,41 17,21 18,61 20,96 21,70
24,48 25,44 29,99 31,37 32,68 33,67 37,95 38,64 39,2 40,12 41,1 43,48 46,2 47,52 48,23
12 0,28 1,93 2,2 3,38 4,8 6,47 9,24 10,8 11,91 13,17 14,59 15,14 16,87 18,15 19,10 20,3
0 22,99 23,97 25,84 26,96 27,58 28,19 29,5 30,63 31,62 33,96 34,26 35,23 36,33 38,9 3
9,68 40,44 41,48 42,43 43,60 45,60 47,92 48,5
13 0,4 1,5 2,30 3,69 4,55 5,70 6,72 8,3 9,26 10,61 11,57 12,17 14,36 15,96 18,77 19,95
21,28 22,38 23,43 25,82 27,38 28,75 33,24 34,91 38,80 40,21 41,36 43,98 44,52 46,87
14 0,3 1,77 2,16 3,21 4,59 7,60 8,57 9,67 10,73 12,59 13,36 16,40 17,87 18,44 21,51 2
```

```
2,84 23,20 24,72 25,73 26,85 28,25 29,80 32,74 33,30 34,65 35,34 36,76 37,50 38,22 43,9
1 46,62 47,69 48,25 49,11
15 0,9 1,89 3,55 5,99 6,2 7,28 8,83 9,6 10,96 11,84 12,14 13,96 16,49 17,75 18,59 19,15
20,21 21,13 22,62 23,42 24,45 25,81 29,76 33,39 34,75 35,24 37,55 38,8 39,93 40,67 42,5
6 43,30 45,43 46,54 47,21 48,85
16 0,1 5,54 6,8 8,45 9,84 11,41 12,87 14,40 15,49 19,98 20,6 21,51 22,68 24,77 25,81 2
6,21 28,1 29,68 30,66 32,40 33,23 34,83 35,80 37,85 38,13 41,24 42,16 43,32 44,38 45,12
47,41 49,87
17 2,45 3,79 5,14 6,37 8,63 9,3 10,52 11,21 14,87 15,75 18,0 19,40 21,40 22,88 24,40 2
5,30 26,62 27,5 28,20 30,33 32,33 35,71 36,35 37,30 38,83 40,17 42,41 43,97 44,81 45,28
46,8 47,69 48,42 49,66
18 0,40 1,7 2,71 3,58 4,76 5,32 6,48 7,51 11,61 12,15 13,77 14,44 15,59 17,0 20,81 21,7
5 22,25 23,28 24,42 25,73 26,23 27,78 29,29 30,49 31,2 32,76 33,0 35,97 36,83 37,63 4
0,39 41,29 42,47 43,88 45,37 47,22 48,97 49,88
19 1,61 4,16 7,58 8,85 10,71 12,10 13,95 15,15 16,98 17,40 20,44 24,30 25,12 26,16 27,2
2 30,76 33,20 34,22 35,82 37,26 39,33 40,94 41,48 42,95 43,69 45,77 47,90 48,57 49,23
20 0,40 1,7 2,45 4,77 6,31 7,1 9,96 11,96 12,30 15,21 16,6 18,81 19,44 22,73 23,52 25,3
8 27,3 28,1 30,13 31,26 32,26 35,34 36,90 37,66 41,32 42,14 44,20 46,91 48,4
21 0,21 1,45 3,0 4,38 5,4 6,52 7,2 9,83 10,3 11,70 13,28 14,51 15,13 16,51 17,40 18,75
22,37 25,75 27,25 33,67 34,52 35,78 36,29 38,89 39,0 41,94 42,41 44,75 45,16 47,2 49,11
22 1,86 2,91 3,75 4,72 5,19 8,6 9,43 10,62 12,99 13,38 14,84 15,62 16,68 17,88 18,25 2
0,73 21,37 24,91 26,17 27,29 28,87 29,37 30,83 31,99 32,9 33,49 34,41 36,28 39,68 41,22
42,56 45,47 46,20 48,39 49,19
23 4,17 5,47 6,69 12,97 13,43 14,20 15,42 18,28 20,52 25,7 26,22 27,80 29,15 30,41 31,4
6 32,45 35,83 36,22 38,8 39,60 43,25 44,76 46,79 47,93 48,2
24 0,28 1,98 2,70 3,89 4,48 5,35 6,70 9,28 10,11 11,48 14,72 15,45 16,77 17,40 18,42 1
9,30 22,91 25,43 26,8 27,91 29,60 31,65 32,48 35,80 36,40 37,36 38,48 39,1 41,72 42,85
43,58 45,11 46,23 47,11 48,10 49,47
25 0,16 1,84 2,13 3,35 5,17 8,59 11,44 12,84 13,82 14,73 15,81 16,81 17,30 18,73 19,12
20,38 21,75 23,7 24,43 26,93 27,77 28,58 31,88 33,73 34,80 35,49 36,25 37,39 38,72 39,2
40,37 42,84 44,12 45,63 46,17 49,54
26 0,0 1,10 3,30 5,43 6,87 7,57 9,61 10,84 12,96 14,85 16,21 17,62 18,23 19,16 22,17 2
3,22 24,8 25,93 27,12 30,62 31,39 32,38 33,40 34,39 36,37 37,46 40,86 42,88 45,78 46,26
47,95 48,13
27 0,22 1,21 2,94 4,58 6,27 8,47 9,24 10,50 12,58 13,38 17,5 18,78 19,22 20,3 21,25 2
2,29 23,80 24,91 25,77 26,12 28,72 29,41 30,3 33,50 35,58 38,98 41,84 43,84 45,10 47,44
48,68 49,52
28 0,19 3,51 5,45 7,86 9,54 12,19 13,75 14,25 16,1 17,20 20,1 22,87 25,58 27,72 29,25 3
0,59 31,46 32,28 33,59 34,28 36,90 37,54 41,29 42,2 43,85 47,11 48,92
29 1,75 2,32 3,80 4,5 5,85 7,91 9,4 10,52 11,99 12,5 14,80 15,76 16,68 18,29 22,37 23,1
5 24,60 27,41 28,25 32,73 33,19 35,14 37,22 40,32 41,36 42,35 43,69 44,79 45,71 46,16 4
9,8
30 1,9 4,87 5,36 6,8 8,77 9,87 10,16 12,63 16,66 17,33 18,49 19,76 20,13 22,83 23,41 2
6,62 27,3 28,59 31,51 32,98 33,68 35,26 36,77 37,88 38,44 40,55 41,15 42,34 43,80 46,93
47,40 48,17
31 1,13 2,58 4,75 6,92 10,73 11,37 12,62 18,2 20,26 22,99 23,46 24,65 25,88 26,39 28,46
30,51 32,74 33,6 34,86 36,20 37,56 38,20 40,91 41,87 42,78 43,51 44,89 46,16 47,12 49,3
1
32 1,34 2,59 3,69 4,8 5,46 6,54 9,50 11,68 14,74 16,40 17,33 18,76 20,26 22,9 23,45 2
4,48 26,38 28,28 29,73 30,98 31,74 37,12 38,47 39,23 40,18 41,92 42,61 43,54 44,25 45,6
```

```
0 47,73 49,8
33 0,0 1,2 2,44 3,34 6,84 9,24 10,61 11,67 12,96 13,24 14,30 15,39 16,23 18,0 19,20 2
1,67 22,49 25,73 26,40 27,50 28,59 29,19 30,68 31,6 37,80 38,45 40,3 41,81 43,30 44,3 4
5,98 46,41 47,20 49,88
34 1,98 2,43 4,45 5,97 6,33 9,37 10,49 12,26 13,91 14,65 15,75 16,83 19,22 21,52 22,41
25,80 26,39 28,28 31,86 35,66 36,2 39,85 42,18 43,57 44,80 45,21 46,60 48,64
35 0,23 1,33 2,38 3,6 6,31 7,96 8,86 9,46 10,32 12,23 14,34 15,24 16,80 17,71 18,97 1
9,82 20,34 21,78 23,83 24,80 25,49 27,58 29,14 30,26 34,66 36,32 37,27 38,25 39,88 41,8
3 42,40 44,90 46,92 47,68 48,9 49,6
36 0,18 2,38 4,31 7,41 9,5 10,57 12,33 14,76 17,35 18,83 20,90 21,29 22,28 23,22 24,40
25,25 26,37 28,90 30,77 31,20 34,2 35,32 38,33 41,12 42,53 44,65 47,89 48,59 49,26
37 0,5 1,67 4,14 6,74 7,43 8,26 9,78 10,8 11,95 14,50 15,55 16,85 17,30 18,63 19,26 2
0,66 24,36 25,39 26,46 28,54 29,22 30,88 31,56 32,12 33,80 35,27 38,28 39,36 41,50 43,5
2 44,30 45,49 47,79 49,39
38 1,59 2,53 5,46 6,70 7,80 9,89 11,64 12,9 13,80 14,22 15,8 16,13 17,83 21,89 23,8 2
4,48 25,72 27,98 30,44 31,20 32,47 33,45 35,25 36,33 37,28 40,86 42,3 43,81 44,38 45,30
46,39 49,77
39 0,9 1,73 2,31 4,55 6,10 8,11 9,27 10,25 11,2 12,68 15,93 19,33 21,0 22,68 23,60 24,1
25,2 32,23 34,85 35,88 37,36 42,22 43,19 44,87 45,44 49,40
40 1,97 3,74 6,59 9,19 10,66 11,12 12,44 13,21 15,67 17,17 18,39 19,94 25,37 26,86 29,3
2 30,55 31,91 32,18 33,3 38,86 41,77 42,28 43,78 44,60 45,45 47,69 49,56
41 1,46 2,33 3,53 5,43 6,13 7,92 10,38 11,1 12,48 13,36 16,24 18,29 19,48 20,32 21,94 2
2,22 24,72 27,84 28,29 29,36 30,15 31,87 32,92 33,81 35,83 36,12 37,50 40,77 43,84 44,5
5 46,75 49,0
42 0,46 2,4 3,68 4,16 6,80 8,92 9,34 12,43 15,56 16,16 17,41 18,47 19,95 20,14 21,41 2
2,56 24,85 25,84 26,88 28,2 29,35 30,34 31,78 32,61 34,18 35,40 36,53 38,3 39,22 40,28
43,19 44,75 46,99 48,81 49,78
43 0,10 1,58 4,66 5,6 6,91 7,40 8,99 11,48 12,60 13,98 14,91 15,30 16,32 17,97 18,88 1
9,69 23,25 24,58 27,84 28,85 29,69 30,80 31,51 32,54 33,30 34,57 37,52 38,81 39,19 40,7
8 41,84 42,19 45,62 46,10 47,6 49,11
44 0,74 1,47 3,13 4,13 5,85 13,52 16,38 17,81 20,20 21,75 23,76 25,12 29,79 31,89 32,25
33,3 34,80 35,90 36,65 37,30 38,38 39,87 40,60 41,55 42,75 45,68 46,76 47,67 48,65 49,8
7
45 0,77 1,11 2,85 3,9 4,21 5,4 6,3 7,70 9,35 12,60 15,43 16,12 17,28 18,37 19,77 21,16
22,47 24,11 25,63 26,78 27,10 29,71 32,60 33,98 34,21 37,49 38,30 39,44 40,45 43,62 4
4,68 46,27 47,74 48,55
46 2,29 3,74 4,49 7,35 8,45 9,80 11,2 13,87 14,62 15,54 17,8 20,91 22,20 23,79 24,23 2
5,17 26,26 29,16 30,93 31,16 33,41 34,60 35,92 38,39 41,75 42,99 43,10 44,76 45,27 47,8
8 48,91
47 0,29 1,31 3,56 5,98 7,93 10,84 11,52 12,92 14,69 15,21 16,41 17,69 18,22 19,90 21,2
23,93 24,11 26,95 27,44 28,11 30,40 31,12 32,73 33,20 35,68 36,89 37,79 40,69 43,6 44,6
7 45,74 46,88 48,43
48 1,23 3,89 6,15 7,50 9,87 10,20 11,23 12,5 14,25 15,85 17,42 18,97 19,57 20,4 22,39 2
3,2 24,10 26,13 27,68 28,92 30,17 34,64 35,9 36,59 42,81 44,65 45,55 46,91 47,43
49 0,88 1,4 2,88 3,76 6,35 7,98 8,51 10,59 14,11 16,87 17,66 18,88 19,23 21,11 22,19 2
4,47 25,54 27,52 29,8 31,31 32,8 33,88 35,6 36,26 37,39 38,77 39,40 40,56 41,0 42,78 4
3,11 44,87
```

This comment was left by Joncody J Wood:

"Thanks for putting together these test cases! They contain a gotcha for developers, though. These data samples are enforcing the directionless nature of the graph through data duplication (eg: 0->1 = 37 & 1->0 = 37), which is not something you will encounter in the assignment dataset. Each edge is stated once, and the tracking the bi-directional relationship is up to the implementation.

8 · flag

 Pavel Velikhov · a month ago

Here is a test case from the class example:

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

the answer is (only costs):
{1: 0, 2: 1, 3: 3, 4: 6}

Here's the same example with a tiny non-connected graph, just to make sure you got it right:
1 2,1 3,4
2 3,2 4,5
3 4,3
6 7,1

the answer is (only costs):
{1: 0, 2: 1, 3: 3, 4: 6, 6: 1000000, 7: 1000000}

3 · flag

 Stefano · a month ago

Wouldn't this require a selection of the connected components of the graph (which is not part of this problem)? Otherwise how can you stop the while loop?
In fact the loop compares the dimension (how many vertexes inside it) of the graph, in this case the connected part of it, and the processed subset of it; in this case the subset would never reach the dimension of the graph since it is not possible to include non-connected vertexes inside said subset (e.g. you never process non-connected components).

0 · flag

 Pavel Velikhov · 19 days ago

No, there is no need to compute connected components. There will be no edges from the "source" part of the graph to the "disconnected" part. So the algorithm will never discover there vertices.

0 · flag

+ Comment

Dmitrii Petukhov · a month ago

Test cases are really nice, but I'm not sure about this part of Joncody J Wood comment:

"*These data samples are enforcing the directionless nature of the graph through data duplication (eg: 0->1 = 37 & 1->0 = 37), which is not something you will encounter in the assignment dataset*"

I don't think it applies to our text file, I'm pretty sure it's directionless through data duplication .

0 · flag

Pavel Velikhov · a month ago

This applies to only the last two cases given previously

0 · flag

+ Comment

Jon Hoffman · 25 days ago

This problem really gave me some headaches. I can post some more tips later, but for now just a few.

The test cases above are not named (!?) making it difficult to refer to them, so I will call the 15-node test case "testcase_15" and the 50 node test case "testcase_50".

testcase_50 has the following output:

Starting vertex = 1
Distances from vertex 1 to nodes 0 through 49:
2 0 10 7 14 11 11 6 8 5
10 5 12 5 5 11 3 2 2 18
5 7 14 11 8 9 2 7 4 9
9 4 12 2 12 10 10 7 9 7
5 4 6 12 5 11 7 9 9 4

testcase_15 has the following output:

Starting vertex = 1
Distances from vertex 1 to nodes 0 through 14:
22 0 26 24 30 16 28 16 46 20
28 15 16 33 25

(somebody please verify both of these!!)

The above was printed with the following code snippet (c++):

```
std::cout << "Starting vertex = " << s << "\n";
std::cout << "Distances from vertex " << s << " to nodes 0 through " << X.size()-1 << ":\n";
```

```
for (int ii = 0; ii   {
    std::cout<< A[ii] << " ";
    if ((ii+1)%10 == 0)
        std::cout<< "\n"; //print return after every 10th value.
}
```

I got really tired of hardcoding the PA5 output every time (commenting, uncommenting, etc) so I finally wrote this:

```
if (fn==PA5fn and s==1) //fn is the file name used for this run and PA5fn is the PA5 file name.
{
    std::cout << "\n---------PA5 answer: \n";
    std::vector<int> PA5 = {7,37,59,82,99,115,133,165,188,197};
    for (uint kk = 0; kk < PA5.size(); ++kk)
        std::cout << A[PA5[kk]] << ",";
    std::cout << "\n\n";
}
```

which will output exactly the text for the final answer, plus a bonus comma.

EDIT I don't know what the deal is, but every time I try to paste the code it gets somewhat corrupted. I give up.

↑ 1 ↓ · flag

Phillip Henry · 20 days ago

Hi, Jon.

I can confirm I also got the answers you stated for the 15- and 50-vertex problem.

Phillip

↑ 1 ↓ · flag

Jon Hoffman · 20 days ago

OK, thanks Phillip. I always feel uneasy after posting unverified solutions, I don't want to steer somebody in the wrong direction.

↑ 0 ↓ · flag

cheng-chih yang · 20 days ago

Hi Phillip, Jon,

Mine is the same as yours too.

↑ 0 ↓ · flag

+ Comment

Jenny · 19 days ago

Why are all the given test cases are directed graphs when the problem set spec is an undirected graph?

0 · flag

Pavel Velikhov · 19 days ago

Good question. The graph in the programming assignment is a directed graph with symmetric edges. E.g. for vertex 1 we have an edge to vertex 80 with weight 982, and for vertex 80 we have an edge to vertex 1 with weight 982. In general, every undirected graph can be converted to a directed graph with the symmetry option and you will get the same result from running the algorithm on it.

0 · flag

+ Comment

Stefan Iustin Oprean · 19 days ago

I would add a test case that made me review the code:

input

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

output:

```
1 0 []
2 1 [2]
3 2 [2, 3]
4 3 [2, 3, 4]
5 4 [2, 3, 4, 5]
6 4 [8, 7, 6]
7 3 [8, 7]
8 2 [8]
```

7 · flag

Phillip Henry · 19 days ago

Excellent submission, Stefan. It forced me to review my code too. I was keeping track of which nodes I had visited not which connections I had seen. Thanks!

2 · flag

Juan Pablo Valencia Gómez · 5 days ago

Saved my life man, thanks!

1 · flag

+ Comment

 Vincent Su · 17 days ago

Hi,

My code passes all the test cases, and it seems to work fine on the assignment file. But when I submitted it the system tells me it's not correct. I submitted the results in this format: "1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000, 1000"

Any help is much appreciated.

Thanks,
Vincent

0 · flag

 Vincent Su · 17 days ago

Found out what the problem was... cannot have space after each comma. The format must be "1000,1000,1000...."

1 · flag

+ Comment

 Nima Farnoodian · 17 days ago

Hi folks, I've got a problem with putting answer. I've sent my answer twice, and each time I got no score. I tested my algorithm on several fictitious graphs and the results were the same as expected. So I think something goes wrong here. Now I wanna know why it happens. Whether It might be due to decoding text file or due to putting answers. If you did successfully, please tell that how the format must be (for example the format of answer must be between two quotes like "1000,1000,1000,1000,1000,1000,1000,1000,1000,1000" or no quot needed for answer like 1000,1000,1000,1000,1000,1000,1000,1000,1000,1000), and what you got as the distance of

two vertices (e.g. the distance of vertex 1 to vertex 20 or the distance of vertex 1 to vertex 150). Frankly, I wanna compare the distances from vertex 1 to some other vertices (those are not answers of PA5) that my algorithm gives as output with yours for finding a solution for the problem. It's my third chance to put answer, I must be careful about that.
I really appreciate you in advance.

0 · flag

 Vincent Su · 17 days ago

Hi,

I had the format problem earlier. The correct format is:
1000,1000,1000,1000,1000,1000,1000,1000,1000,1000

There should be no quotes, and no space in between the numbers and the commas.

Hope this helps!

0 · flag

Anonymous · 17 days ago

thanks for sending that.

0 · flag

 Jingwen Ouyang · 15 days ago

i got no score in the beginning because i had space in between, now i past! :) thanks for clarifying it!

0 · flag

+ Comment

Anton Gladky · 17 days ago

Same here, all test cases are passed, but for the real task I get 0 scores. Can anyone please send some more test cases? Thanks

0 · flag

 Vincent Su · 17 days ago

Anton,

Do you think it might be caused by your answer format?

0 · flag

Anton Gladky · 17 days ago

I do not think so. It looks like my distances do not match distances in answer from Jon Hoffmann.

0 · flag

 Nima Farnoodian · 17 days ago

Hi,
If your algorithm dose work well for your test cases, you definitely made mistake in converting text file as what I did. I'm solving that now. I found out why it doesn't work for this data set. Try it again...

0 · flag

 Nima Farnoodian · 17 days ago

I'm sorry, I read your last comment now. I found out that my wrong answer come from some error in my code for decoding text file. Because of that I think if you have the same problem (algorithm works well on test data but doesn't give true answer for dijkstraData), you'd better check your code responsible for converting text file. You might have made some mistake somewhere in your code. Check data and see whether converted data represent the same graph or not. Good Luck!

0 · flag

+ Comment

 Nima Farnoodian · 17 days ago

Yeah! I did that... thank you all...

0 · flag

+ Comment

Ahmed Yossre · 16 days ago

Same issue, for anyone having correct test cases but wrong final solution
I was splitting the edges by ' ' where it should have been split by '\t' check formatting and debug to see your edges if they were added correctly as they look in the dijkstraData file.
happy coding :D

0 · flag

+ Comment

Jeff Beckley · 16 days ago

When I first implemented the version of the algorithm using a heap I had a bug that made it miscalculate just one of the shortest paths of the 200 vertices in the assignment problem, yet all of my test cases (including all the ones given in this thread) were being computed correctly. The PA#5 graph is more complex than most examples, and also includes some edges of distance 0 that may introduce some subtle cases not seen in other graphs.

0 · flag

Jonathan D Johnston · 16 days ago

Not sure what you mean by "edges of distance 0". Each line of dijkstraData.txt consists of:

```
tail  head0,length0 ... headN,lengthN
```

Every length in the file is > 0 (just checked).

0 · flag

Jeff Beckley · 16 days ago

Sorry, it was the n=50 example above that had the zero distances, not the PA#5 graph.

0 · flag

+ Comment

Anton Gladky · 15 days ago

Thanks all for the help and advicesI I finally did it!

0 · flag

+ Comment

Yago Guimarães · 15 days ago

Could anyone gimme some examples of shortest paths from the assignment graph or more test cases? I'm passing all test cases but no luck with the submissions, and I have only one left...

0 · flag

Heaven Hodges · 15 days ago

Me too. It's frustrating.

0 · flag

Ahmed Yossre · 15 days ago

double, trible check that your code reads the dijdata.txt correctly, note that the spaces are '\t' not ' ' check that it reads the edges correctly

0 · flag

Jeff Beckley · 15 days ago

Here's the minimum paths of the first few vertices of the PA#5 graph:

```
1: 0 1
2: 2971  1 -(508)-> 114 -(168)-> 129 -(517)-> 85 -(1227)-> 53 -(236)-> 34 -(315)-> 2
3: 2644  1 -(982)-> 80 -(108)-> 119 -(617)-> 195 -(652)-> 144 -(285)-> 3
```

```
4: 3056  1 -(2437)-> 49 -(619)-> 4
```

The first number (before the colon) is the finish vertex #. The second number is the total minimum distance from vertex #1 to that vertex. And the last part is the path to get that minimum distance, where the numbers outside parentheses are the vertex #s and the numbers in parentheses are the distances between those 2 vertices. So for example, for the minimum path between vertex #1 and vertex #4, you go from vertex #1 to vertex #49 (which has an edge length/distance of 2437), and then you go from vertex #49 to vertex #4 (which has an edge length/distance of 619), which gives you a total path distance of 3056 (2437 + 619).

(note that none of these vertices above are ones that you should report in your solution to PA#5, since the assignment asks you to report the minimum distances to vertices #7, #37, #59, ...)

↑ 1 ↓ · flag

+ Comment

New post

To ensure a positive and productive discussion, please read our forum posting policies before posting.

B I Link <code> Pic Math Edit: Rich Preview

Make this post anonymous to other students

Subscribe to this thread at the same time

Add post