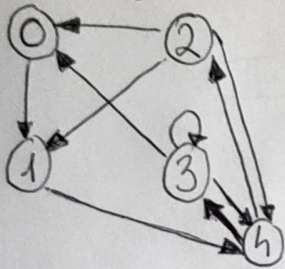


practical work no 2

2. write a program that, given a directed graph and two vertices, finds a lowest length path between them, by using a backward breadth-first search from the ending vertex.

we're using vertex 1 as the starting vertex and vertex 2 as the ending vertex

random_graph_bbfs_1.txt



predecessors dictionary:

Keys	0	1	2	3	4
	↓	↓	↓	↓	↓
value	[2 3]	[0 2]	[4 2]	[3 4]	[1 2 3]

vertex 1 = 1 vertex 2 = 0	current_node	next_node	queue	visited	child parent
initialization			← 0 ←	{0}	0 1 2 3 4
iteration 1	0		← 2 ←	{0, 2}	0 1 2 3 4
iteration 1.1		2	← 2 ←	{0, 2}	0 1 2 3 4
iteration 1.2		3	← 2 3 ←	{0, 2, 3}	0 1 2 3 4
iteration 2	2		← 3 ←	{0, 2, 3}	0 1 2 3 4
iteration 2.1		4	← 3 4 ←	{0, 2, 3, 4}	0 1 2 3 4
iteration 3	3		← 4 ←		0 1 2 3 4
iteration 3.1		3			0 1 2 3 4
iteration 3.2					0 1 2 3 4
iteration 4	4		← 1 ←	{0, 1, 2, 3, 4}	0 1 2 3 4
iteration 4.1		1	← 1 ←	{0, 1, 2, 3, 4}	0 1 2 3 4

• next_node = vertex 1 = 1 ⇒ BREAK (path_found = True)

• the path is built from child

4	0	0	2
---	---	---	---

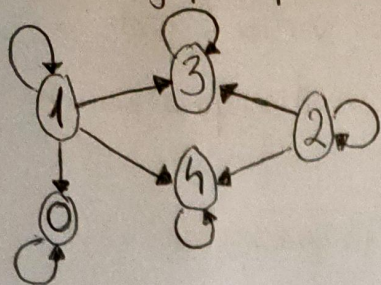
 beginning with vertex 1 = 1

vertex 1 = 1 → child[1] = 4 → child[4] = 2 → child[2] = 0 → child[0] = None

• path = [1, 4, 2, 0]

• length = len(path) - 1 = 3

random-graph-bbfs-2.txt



predecessors dictionary

Keys	0	1	2	3	4
value	$\begin{bmatrix} 0 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1 \end{bmatrix}$	$\begin{bmatrix} 2 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$	$\begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix}$

vertex ₁ = 3 vertex ₂ = 4	current - node	next - node	queue	visited	child										
initialization			← 4 ←	{4}	<table><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td></td><td></td><td></td><td></td><td style="background-color: #cccccc;"></td></tr></table>	0	1	2	3	4					
0	1	2	3	4											
iteration 1	4		← ←												
iteration 1.1		1	← 1 ←	{1, 4}	<table><tr><td></td><td>4</td><td></td><td></td><td style="background-color: #cccccc;"></td></tr></table>		4								
	4														
iteration 1.2		2	← 1 2 ←	{1, 2, 4}	<table><tr><td></td><td>4</td><td>4</td><td></td><td style="background-color: #cccccc;"></td></tr></table>		4	4							
	4	4													
iteration 1.3		4													
iteration 2	1		← 2 ←												
iteration 2.1		1													
iteration 3	2		← ←												
iteration 3.1		2													

• queue is empty \Rightarrow path not found \Rightarrow there's no path ^{from} vertex 3 ^{to} vertex 4

1 \longrightarrow 100

100 \longrightarrow 1

graph 1K.txt

1 \longrightarrow 5 \longrightarrow 649 \longrightarrow 453 \longrightarrow 107 \longrightarrow 647 \longrightarrow 100

100 \longrightarrow 416 \longrightarrow 354 \longrightarrow 865 \longrightarrow 109 \longrightarrow 1

6

5

graph 10K.txt

1 \longrightarrow 7317 \longrightarrow 4118 \longrightarrow 2604 \longrightarrow 690 \longrightarrow 1494 \longrightarrow 739 \longrightarrow 4722 \longrightarrow 100

100 \longrightarrow 5568 \longrightarrow 9908 \longrightarrow 1820 \longrightarrow 5308 \longrightarrow 528 \longrightarrow 4260 \longrightarrow 1

8

7

graph 100K.txt

1 \longrightarrow 17024 \longrightarrow 27471 \longrightarrow 14969 \longrightarrow 3075 \longrightarrow 40733 \longrightarrow 85480 \longrightarrow 14973 \longrightarrow 100

100 \longrightarrow 44340 \longrightarrow 54527 \longrightarrow 6606 \longrightarrow 53263 \longrightarrow 95930 \longrightarrow 98655 \longrightarrow 58288 \longrightarrow 1

8

8