Adjacency Lists

```
0:  1,  4
1:  0,  2
2:  1,  3,  5,  6
3:  2
4:  0,  5
5:  2,  4,  6,  7
6:  2,  5
7:  5,  9
8:  9
9:  7,  8
```

In this question, you are going to do a Breadth-First-Search starting at node 0. I want you to do this like I did in class, maintaining a queue, a distance vector and a back-link vector. Here are the things that I want you to enter:

Question 1: In what order are the nodes visited? Enter this as a single 10-digit string with no spaces.

Question 2: What is the distance vector? Enter this as a single 10-digit string, where the character in digit *i* is the distance of node *i* from node 0.

Question 3: What is the back-link vector? Enter it as a single 10-digit string, where the character in digit *i* is the id of the node that put node *i* onto the queue. Put a dash '-' for node 0.

Answers to the clicker questions

Build this up like I have done in class with the other BFS examples:

```
Node  Action           Distance      Back-Links    Queue
                       0123456789    0123456789    (push_back, pop_front())
      Start            0---------    ----------    0
0     Push 1 & 4       01--1-----    -0--0-----    1, 4
1     Push 2           012-1-----    -01-0-----    4, 2
4     Push 5           012-12----    -01-04----    2, 5
2     Push 3, 6        0123123---    -012042---    5, 3, 6
5     Push 7           01231233--    -0120425--    3, 6, 7
3     Nothing          01231233--    -0120425--    6, 7
6     Nothing          01231233--    -0120425--    7
7     Push 9           01231233-4    -0120425-7    9
9     Push 8           0123123354    -012042597    8
8     Nothing          0123123354    -012042597
```

So:

Question 1: 0142536798
Question 2: 0123123354
Question 3: -012042597