

Chapter 5, Dynamic Programming:

- ▶ Introduction and sample problems (previous week Wednesday)
- ▶ Principles of DP (previous week Friday)
- ▶ DP algorithms, solutions to sample problems (last week Monday)
- ▶ Introduce optimal BSTs / review for test 2 (last week Wednesday)
- ▶ **Test 2**, *not* covering DP (last week Friday)
- ▶ Test 2 retrospective (**Today**)
- ▶ Finish up optimal BSTs (Wednesday)
- ▶ [Begin hash tables (Friday)]

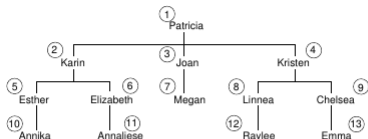
Today:

- ▶ How I scored tests
- ▶ Background of the chutes and ladders problem
- ▶ Comments on other problems
- ▶ Looking ahead

...before asking about the following generation. Consider the family and interviews going deep down branches versus broad across generations:

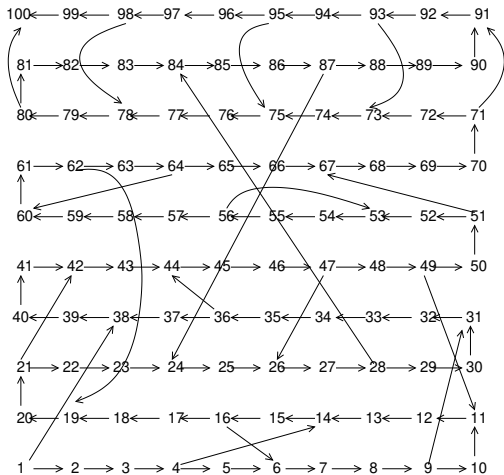
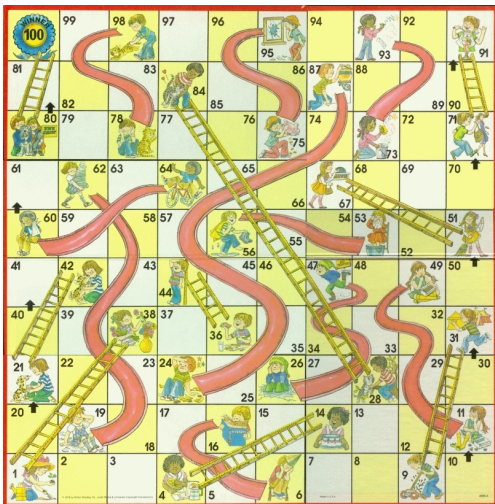


Who is the first ancestor? Patricia Does Patricia have any children? (y/n) y
 Who is the next child? Karin Does Karin have any children? (y/n) y
 Who is the next child? Esther Does Esther have any children? (y/n) y
 Who is the next child? Annika Does Annika have any children? (y/n) n
 Does Esther have any more children? (y/n) n
 Does Karin have any more children? (y/n) y Who is the next child? Elizabeth
 Does Elizabeth have any children? (y/n) y Who is the next child? Annie
 Does Annie have any children? (y/n) n
 Does Elizabeth have any more children? (y/n) n
 Does Karin have any more children? (y/n) n
 Does Patricia have any more children? (y/n) y Who is the next child? Joan
 Does Joan have any children? (y/n) y Who is the next child? Megan
 Does Megan have any children? (y/n) n
 Does Joan have any more children? (y/n) n
 Does Patricia have any more children? (y/n) y Who is the next child? Kristen
 Does Kristen have any children? (y/n) y Who is the next child? Linnea
 Does Linnea have any children? (y/n) y Who is the next child? Raylee
 Does Raylee have any children? (y/n) n
 Does Linnea have any more children? (y/n) n
 Does Kristen have any more children? (y/n) y Who is the next child? Chelsea
 Does Chelsea have any children? (y/n) y Who is the next child? Emma
 Does Emma have any children? (y/n) n
 Does Chelsea have any more children? (y/n) n
 Does Kristen have any more children? (y/n) n
 Does Patricia have any more children? (y/n) n



Who is the first ancestor? Patricia
 How many children does Patricia have? 3
 Who is Patricia's next child? Karin
 Who is Patricia's next child? Joan
 Who is Patricia's next child? Kristen
 How many children does Karin have? 2
 Who is Karin's next child? Esther
 Who is Karin's next child? Elizabeth
 How many children does Joan have? 1
 Who is Joan's next child? Megan
 How many children does Kristen have? 2
 Who is Kristen's next child? Linnea
 Who is Kristen's next child? Chelsea
 How many children does Esther have? 1
 Who is Esther's next child? Annika
 How many children does Elizabeth have? 1
 Who is Elizabeth's next child? Annie
 How many children does Megan have? 0
 How many children does Linnea have? 1
 Who is Linnea's next child? Raylee
 How many children does Chelsea have? 1
 Who is Chelsea's next child? Emma
 How many children does Annika have? 0
 How many children does Annie have? 0
 How many children does Raylee have? 0
 How many children does Emma have? 0

The point is not that there is a right way or wrong way to do it, but that these two interviews are done by essentially



Package Explorer

- JRE System Library [java-1.8.0]
- graph-py [CSCI345_DSA_F22 m...]
 - graph
 - __init__.py
 - graph.py
 - mst.py
 - priorityqueue.py
 - sssp.py
 - traversal.py
 - graph-traversal [CSCI345_DSA...]
 - adt
 - alg
 - BreadthFirstIterator.java
 - BreadthFirstTraversal.java**
 - DepthFirstIterator.java
 - DepthFirstTraversal.java
 - DepthFirstTraversalRecursiv
 - MaxDegree.java
 - PerformOnVertex.java
 - Traversal.java
 - impl
 - test
 - JRE System Library [java-1.8.0]
 - JUnit 4
 - small.graph

RBNode.java VerifyLLRB.java traversal BreadthFirstTraversal.java

```
31
32  /**
33   * Retrieve the distances from the starting point for each vertex
34   * from the most recent traversal done by this object.
35   */
36   public int[] distances() { return distances; }
37
38  /**
39   * Execute a given operation on each vertex of the graph
40   * reachable able from start in a specific ordering
41   * @param g The graph on whose vertices to operate
42   * @param start The vertex from which to start
43   * @param op The encapsulated operation to perform
44   */
45  public int[] traverse(Graph g, int start, PerformOnVertex op) {
46      int[] parents = new int[g.numVertices()];
47      distances = new int[g.numVertices()];
48      for (int i = 0; i < g.numVertices(); i++)
49          parents[i] = distances[i] = -1;
50      distances[start] = 0;
51      parents[start] = start;
52
53      Queue<Integer> worklist = new ListQueue<Integer>();
54      worklist.enqueue(start);
55      distances[start] = 0;
56      while (! worklist.isEmpty()) {
57
58          int front = worklist.remove();
59          op.perform(front);
60          for (int u : g.adjacents(front)) {
61              if (parents[u] == -1) {
62                  worklist.enqueue(u);
63                  distances[u] = distances[front] + 1;
64                  parents[u] = front;
65              }
66          }
67      }
68  }
```

DS&A

▼ Course Options



Materials



Updates



Gradebook



Grade Setup



Mastery



Badges



Attendance



Members



Analytics



Workload Planning

CSCI-345: Section 0 - Data Structures & A ... ▶ Graphs (Ch 4)



Concepts, implementation, and traversal -- Sept 26 & 28

Reading and practice: Sections 4.(1-3)

Due: Friday, September 30, 2022 at 11:59 pm

Read Sections 4.(1-3), pg 265-302. This is a big chunk---spread it out! I'm putting this, together with the exercises, as "due" on Friday, but you should work on it in parallel to what we're doing in class, with the Friday due date being for any pieces you still need to finish up.

Do Exercise 4.(26-29) [pg 360-361]. This is a programming assignment, kind of like a mini-project. You will find starter code in the repository under practice/traversal, and a solution under practice/traversal-soln.

Posted Wed Aug 17, 2022 at 2:00 pm

Comments

There are no comments

Post

Quiz

Indicate which algorithm uses which ADT

Breadth-first traversal

Depth-first traversal

Stack

Queue

2. I did Exercises 4.(26-29)	True/False	1 Point	100.0%
4. Indicate which algorithm uses which ADT	Matching	1 Point	100.0%

6. Consider the following code assuming g is a graph with the usual interface.

```
Queue<Integer> q = new SomeQueue<Integer>();
Set<Integer> discovered = new SomeSet<Integer>();
q.enqueue(0);
discovered.add(0);
while (! q.isEmpty()) {
    int u = q.remove();
    // ...do something with u...
    for (int v: g.getAdjacents(u))
        if (! discovered.contains(v)) {
            q.enqueue(v);
            discovered.add(v);
        }
}
```

(2 points each)

a. Does the following perform breadth-first traversal or depth-first traversal of g from vertex 0?

b. What would you change in the above code to transform it into the other (breadth-first traversal vs depth-first traversal)?

c. Which algorithm (breadth-first or depth-first) would you use to find the shortest path between vertices in terms of number of edges (**not** in terms of edge weights)?

Practice problems for test 2



3 / 5 >



129%

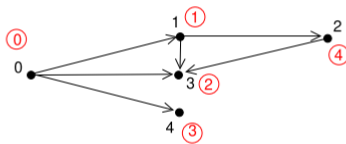


ab]



3. [Graphs.] Recall that *breadth first traversal* is an algorithm pattern used to iterate over all the vertices in a graph that are reachable from a given starting vertex in an order based on the structure of the graph. Specifically, breadth first traversal first visits the source vertex, then all vertices one edge away from the source, then all the vertices two edges away from the source, etc.

For an example, consider the following graph. The circled numbers indicate the order in which they are visited in breadth first traversal starting from 0.



Finish the class `bfi.BreadthFirstIterator`, which iterates over the vertices in the graph in breadth-first order from a given source. Specifically, when a given a graph and a starting vertex `source` are passed to this class's constructor, the resulting iterator behaves such that successive calls to `next()` return the vertices reachable from `source` in breadth-first order. For example, when given the graph above and vertex 0 as `source`, then calls to `next()` return 0, 1, 3, 4, 2, respectively, after which calls to `hasNext()` return false.

Graphs are represented by the same `Graph` interface that was used in class and the projects, with vertices identified by whole numbers. The class used is `AdjListGraph`, but that shouldn't matter.

4. [Graphs.] Suppose you are given a directed graph and a source vertex s . What

QUICK ACCESS

Package [JUnit]

- test2-given
 - q1adt
 - q2graph
 - CheatsAndLadders.java**
 - ChutesAndLaddersBoard.j
 - LinkedQueue.java
 - Queue.java
 - TestCL.java
 - q3bst
 - JRE System Library [java-1.8]
 - JUnit 4
- test2-soln

```
1 package q2graph;
2
3 import java.util.Iterator;
4
5 /**
6  * CheatsAndLadders
7  *
8  * Placeholder for method to find the fewest moves needed
9  * to finish a chutes and ladders board
10 *
11 * CSCI 345
12 * Test 2
13 */
14 public class CheatsAndLadders {
15
16     /**
17      * Find the fewest number of moves needed to get to the end of the
18      * given board.
19      * @param board A Chutes and Ladders board
20      * @return The fewest number of moves needed to get to the finish
21      */
22     public static int fewestMoves(ChutesAndLaddersBoard board) {
23         // What could you use this for?
24         int[] d = new int[board.size()];
25
26         throw new UnsupportedOperationException();
27     }
28 }
```

Coming up:

Catch up on projects. . .

*Due **Mon, Nov 14** (end of day)*

Do Project 6.1.b as a practice problem

Take quiz (on Section 6.4)

*Due **Wed, Nov 16** (end of day)*

Read Section 6.5

(No quiz on Section 6.5)

*Do **Optimal BST** project (suggested by Monday, Nov 21)*

*Due **Fri, Nov 18** (end of day)*

Read Sections 7.(1 & 2)

Take quiz