Assignment 4

CSE 332: Data Abstractions, Spring 2011
University of Washington
May 2, 2011
due: Wednesday, May 11, 2:15 p.m.

Instructions: Create a PDF representation of your answers and submit it via Catalyst CollectIt. You might wish to prepare the file using LaTeX. You may use the source file of this assignment as a starting point if you use LaTeX. The "pdflatex" command installed on most Linux systems is a convenient way to translate the LaTeX source file into a PDF document. You may prefer to draw the diagrams by hand and scan them to include in your document. The LaTeX source file shows how to include an image, such as a PNG, GIF, or JPG image, in your document.

Be sure that your name is clearly visible at or near the top of the first page. The LaTeX source file also shows where you can put that. The due time for this assignment is 2:15 PM, which means that if you turn it in at the last minute (but this is NOT recommended) from somewhere on campus, you'll still have time to get to class.

1. (15 points)

   John is responsible for keeping the recent-fiction shelf in his club's library sorted by author. About once a week, a reader picks out a couple of books and puts them back, but mixes them up, putting book B1 where B2 should go, and vice-versa. John is diligent and checks for this every day.

   (a) (5 points) Assuming it never happens more than once a day, how can he most efficiently detect when this has occurred and fix it?

   (b) (10 points) When it does occur, on average, how many comparisons between pairs of author names does he have to make? (Assume all books have differently named authors. And assume all books are equally likely to be chosen by readers.)

2. (20 points)

   In items (a) through (d) below, give the requested details about how to sort a collection of playing cards (not necessarily an even 52-card deck – possibly fewer, possibly more) into "Bridge order" by first separating the cards into 4 piles by suit, and then concatenating the piles and separating them into 13 piles by rank and then concatenating them.

   (a) (2 points) How is the concatenation performed?

   (b) (3 points) What detail of the second separation phase is particularly important for this to work?

(c) (2 points) What general kind of sorting algorithm is this (Heapsort, Mergesort, etc.) and does it use comparisons?

(d) (3 points) What is the "running time" of this method as a function of the number of cards, number of ranks and number of suits?

(e) (10 points) Demonstrate the sorting algorithm on the following sequence of 10 cards.

`5D, AS, JH, 2C, 10D, QH, 7S, 3S, 5C, KH.`

Show the following 4 states of execution: (i) after distribution by suits (show the 4 piles). (ii) after concatenation following (i). (iii) after distribution by ranks (show 13 piles, even though some may be empty). (iv) after the final concatenation.

3. (25 points) An undirected graph G has two parts: a set V of vertices and a set E of edges. Each edge can be considered as an unordered pair of elements of V.

(a) (5 points) Give the adjacency list rep. of the undirected graph G_a that is shown in Figure 1. (Use a similar style of adjacency list presentation as that used in part (b) below.)

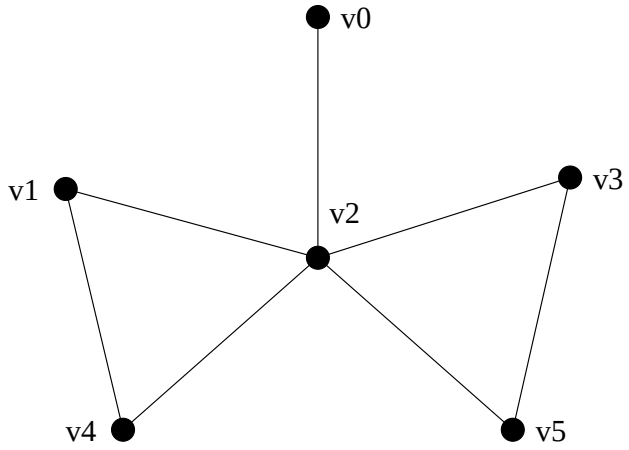

Figure 1: The undirected graph G_a for problem 3a.

(b) (3 points) Draw the undirected graph whose adjacency lists are the following.

```
Ann: Bob, Cal, Deb, Hal;
Bob: Ann, Cal, Eby, Fil, Hal;
Cal: Ann, Bob, Deb, Eby;
Deb: Ann, Cal, Eby, Gus, Hal;
Eby: Bob, Cal, Deb, Fil, Gus;
Fil: Bob, Eby, Gus, Hal;
Gus: Deb, Eby, Fil, Hal;
Hal: Ann, Bob, Deb, Fil, Gus;
```

Note that this graph can be drawn on paper such that no two edges cross each other. If you succeed in drawing it that way, you'll get 2 extra points. (Hint: space the vertices evenly around a circle with Ann at the top.)

(c) (5 points) Consider the graph shown in Figure 2. Use breadth-first search from John to label all nodes with their distance from John. (where distance = minimum number of edges in a path from A to B). You may show the results either by putting an extra set of labels on the graph (with circles around them) or by giving a table of the vertices (in alphabetical order) with their distances.

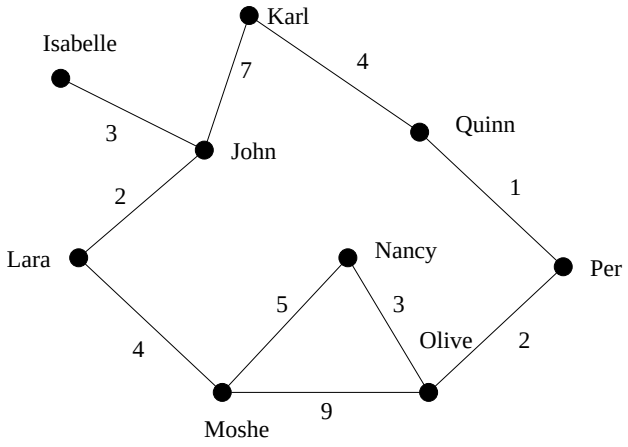

Figure 2: The undirected graph G_a for problem 3c.

(d) (10 points) Use Dijkstra's algorithm to label all nodes with their distance from John (where distance(A,B) = min path length from A to B, and path length is the sum of the edge distances along the path.) You may show the results either by putting an extra set of labels on the graph (with squares around them) or by giving a table of the vertices (in alphabetical order) with their distances.

4. (10 points)

Give a Java code excerpt for creating and running two threads, where one thread counts normally from 1 to 10, and the other thread counts down from -1 to -10. Whenever a count value is changed, it should be printed out immediately. The main thread should wait for the two counting threads to finish. To do this, you should subclass Thread, creating a new class called CountingThread, and your main method should create two instances of CountingThread.

5. (10 points) This problem is about fork-join parallelism.

(a) (5 points) Give pseudocode for a fork-join parallel program that uses P processors and computes the dot product of two vectors V_1 and V_2, each of length N where $N > P$.

(b) (5 points) Give a Java implementation of the above. Make use of the ForkJoin framework, as described in the handout by Grossman.

(c) (Optional for 10 points of extra credit). Actually debug your implementation and paste in the Java source code, and the transcript of a brief test run here.