CS310: Advanced Data Structures and Algorithmns

Spring 2014 Assignment 5

Due: Tuesday, April 29 2014, in class

Goals

Prefix Codes, Character codes, and Intro to Graphs

Questions

1. Start from Weiss, pg. 500, problem 12.1:

   (a) To make the tree fully-specified, put the lower-weight subtree on the right on each merge operation, and adjust newlines frequency to 101. Start by listing the weights in increasing order from left to right, with their symbols below them on the next line, leaving space above to build trees. You will find that you need to move the space symbol down to the other end, when it gets chosen.

   (b) What is the resulting binary code for the most frequent symbol? The least frequent?

   (c) With the coding scheme above, code the 7-symbol text "04: 19," . Show the binary string and the bytes in hex.

   (d) With the coding scheme of problem 1, decode 11001110111100000.

   (e) Suppose we want to do fast decoding for this scheme. Show the beginning of the decode array, that is, covering the first three codes, using ... for repetitive entries as in the notes.

2. Here is some text encoded in UTF-8, then printed in hex bytes: 41 42 31 32 c2 a9 28 29

   (a) What is it in UTF-16, used in Java? Write it in hex shorts: 0041 ...

   (b) What symbols does this represent?

3. Consider the program in Figure 4.21, pg. 140 (or its more modern equivalent with Scanner to get lines and

   ```
   line.split(``\\s++'')
   ```

   to tokenize them), and assume the user types in "10 -3" followed by a newline, in ASCII of course (there is a space between 0 and -).

   (a) What are the ASCII codes (in hex) reaching Java?

   (b) What are the Unicode (UTF-16) characters in oneLine, after the conversion?

   (c) What is returned by str.nextInt on line 22? on line 23? (What characters, in hex)

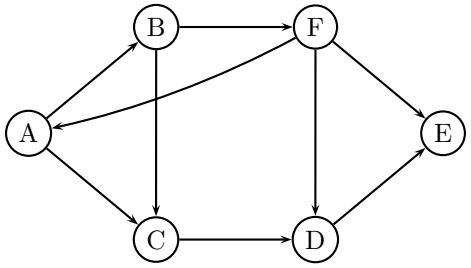
   (d) What is put in x? in y? (in hex)

4. Weiss, pg. 568 Intro to graphs. Read the scenarios in problems 14.21, 22, 25, and 26 (don't try to solve the problems, just read). For each of 14.22, 25, and 26:

   (a) Define what a vertex is for the graph, and give examples. For ex. In 14.21, a vertex is a team. For example, teams A, B, etc.

(b) Define what an edge is, that is, whether directed or undirected, and what vertices it connects, under what circumstances. For ex. in 14.21, an edge is directed, and exists from team A to team B if A beat B in some game.

(c) Show the example graph in the problem. For ex. In 14.21:



(d) Show the adjacency list representation of your example graph, something like Figure 14.2, except that you may need to use the vertex names instead of 0, 1, ... and don't need cost values:

Ex. For 14.21:

A: [B, C]
B: [C, F]
C: [D]
D: [E]
E: [A]
F: [D, E]

5. Consider the graph in Weiss ex. 14.4. Show the in-degree counts for all the nodes at each step of the topological sort algorithm discussed on page 556 and fig. 14.30. Here is a start:

A	B	C	D	E	Out
1	2	0	2	2	C
0	2	0	1	2	A

...

6. For the graph of figure 14.4 (page 532), show the DFS for this graph starting from C, like this:

DFS from C:
visit C before edges
visit edge CA
visit A before edges
visit edge AB
...
visit A after edges
...

and indicate how to find a topological sort from this DFS trace. Notice: To make the graph acyclic reverse the directions of the C-D and the B-D edges.

7. For the graph of exercise 14.5, show the BFS for this graph starting from C, using alphabetical ordering of to-nodes, so that C-A precedes C-B as an edge out of C. Also show the queue of nodes waiting to be visited, and the set of as-yet unseen nodes. Here's a start:

Stage	Queue	Unseen
Start	[C]	{A,B,D,E,F}
visit C	[]	{A,B,D,E,F}
visit edge CA	[A]	{B,D,E,F}
visit edge CB	[A,B]	{D,E,F}
visit A	[B]	{D,E,F}