CS3114 Fall 2008 Homework Assignment 2
Due Tuesday, November 11 at 11pm
50 points

1. Using the weighted union rule and path compression, show the array for the parent pointer implementation that results from the following series of equivalences on a set of objects indexed by the values 0 through 15. Initially, each element in the set should be in a separate equivalence class. When two trees to be merged are the same size, make the root with greater index value be the child of the root with lesser index value.

(2, 3) (4, 5) (6, 5) (3, 5) (1, 0) (7, 8) (1, 8) (3, 8) (9, 10) (11, 14) (11, 10) (12, 13) (11, 13) (14, 1)

2. A full K-ary tree is a tree where every internal node has exactly K children. Find the overhead fraction for a full K-ary tree implementation with space requirements as follows:

1. All nodes store data, K child pointers, and a parent pointer. The data field requires four bytes and each pointer requires four bytes.
2. All nodes store data and K child pointers. The data field requires sixteen bytes and each pointer requires four bytes.
3. All nodes store data and a parent pointer, and internal nodes store K child pointers. The data field requires eight bytes and each pointer requires four bytes.
4. Only leaf nodes store data; only internal nodes store K child pointers. The data field requires four bytes and each pointer requires two bytes.

3. Recall that a sorting algorithm is said to be stable if the original ordering for duplicate keys is preserved. Of the sorting algorithms Insertion Sort, Bubble Sort, Selection Sort, Shellsort, Quicksort, Mergesort, Heapsort, Binsort, and Radix Sort, which of these are stable, and which are not? For each one, describe either why it is or is not stable. If a minor change to the implementation would make it stable, describe the change.

4. Give a permutation for the values 0 through 7 that will cause Quicksort (as implemented in Section 7.5) to have its worst case behavior.

5. Assume that a disk drive is configured as follows. The total storage is approximately 1033MB divided among 15 surfaces. Each surface has 2100 tracks, there are 64 sectors/track, 512 bytes/sector, and 8 sectors/cluster. The disk turns at 7200 rpm. The track-to-track seek time is 3 ms, and the average seek time is 20 ms. Now assume that there is a 512KB file on the disk. On average, how long does it take to read all of the data on the file? Assume that the first track of the file is randomly placed on the disk, that the entire file lies on contiguous tracks, and that the file completely fills each track on which it is found. Show your calculations.