Boğaziçi University
Department of Physics

Phys 499 Spring 2007

Project #1

Due in class Monday, 23 May 2007

During the course of this term project you will study several sort algorithms.

Question 1:
Write a radix sort algorithm in C that sorts n 32-bit unsigned integer numbers stored in the `key` element of the linked-list.

- The prototype of the sort algorithm: `void sort_radix(int n, list_t *list);`,
- Use linked lists for temporary storage "boxes",
- There can be only a constant amount of temporary variables,
- Optimize the code for good run-time.

a) The maximum integer which can be represented by 32-bit integer is about 4 billion, 10 digit number. Use 10 "boxes" for each decimal digit. Test the code for run-time for 1,000,000 random numbers generated between 0 and $(2^{32} - 1)$.

b) Now, think the numbers as binary numbers, and group them as 4-bit digits. This time, you will need only 16 boxes. Repeat the run-time test for the same set of input numbers.

c) Repeat the test for 2-bit groups and single bit groups.

Question 2:
Write a code that uses divide-and-conquer method recursively for arrays of size greater than 4, and invent an efficient piece of algorithm that sorts 4 integers, and use it for the case the array size is less than or equal to 4. Do a run-time test for the same set of input array. Use `void sort_shell(int n, unsigned int *A);` as your prototype.

Question 3:
Repeat the run-time test using the same set of input array for heap-sort. Use `void sort_heap(int n, unsigned int *A);` as your prototype.

Question 4:
Implement the insertion sort using linked lists. Repeat the run-time test using the same set of input array. Use `void sort_insertion(int n, list_t *list);` as your prototype. Note that insertion is $O(1)$ process using the linked lists.

Question 5:
Make a plot of run-times for these 7 cases for different n values. Hint: Since the difference between the run-times may be huge, use (square root of time)-vs-n. Choose your maximum n value appropriately, such that the plot clearly gives an idea about the strengths and drawbacks of each case (it resolves the curves). If necessary, make two separate plots; one for small n values, one for large n value.