Purdue University | Hambrusch
CS 381 | Fall 2008

Assignment 3

Due: Wednesday, September 24, 2008 (before class)

1) (15 pts.) Bob is very disappointed that he won't see new sorting algorithms in CS381. To cheer him up, Alice proposes the following sorting algorithm:

```
Bob-Sort(A,i,j)
    if A[i] > A[j] then swap A[i] and A[j]
    if i + 1 ≥ j then return
    k = ⌊(j − i + 1)/3⌋
    Bob-Sort(A,i,j-k)
    Bob-Sort(A,i+k,j)
    Bob-Sort(A,i,j-k)
```

(i) Alice claims that Bob-Sort(A,1,n) correctly sorts an array of size n. She reasons that sorting the first two thirds, then the last two-thirds, and then again the first two-thirds of an array generates the sorted sequence. Explain why Alice is right.

(ii) Set up the recurrence relation for Bob-Sort and solve it (you can use the Master theorem, if it applies). Is Bob-Sort superior to any of the sorting algorithms mentioned in class?

2) (10 pts.) Consider Algorithm 1 given for the skyline problem. Assume the skyline is represented by a single-linked list with left endpoints arranged by non-decreasing values. The "buildings" are considered by non-decreasing left endpoints. For this scenario give an example of size n on which the algorithm takes $\Theta(n^2)$ time. Show the example for $n = 8$. Explain how it generalizes and how the time bound is achieved.

3) (15 pts.) Give the tightest asymptotic bound you can for $T(n)$ in each recurrence listed below. Assume $T(n) = \Theta(1)$ for $n \leq 2$. If the Master theorem can be used, explain how it is used, what case applies and why. If the Master theorem does not apply, make a proof by induction.

(i) $T(n) = 2T(n/4) + \sqrt{n}$
(ii) $T(n) = 2T(n/2) + n^3$
(iii) $T(n) = T(n/5) + 2T(n/3) + n$
(iv) $T(n) = 6T(n/2) + n \log n$

4) (10 pts.) Review heaps as a data structurew.

(i) Consider a min-heap built on a set of size n. At how many locations can the third smallest element be? At how many locations can the largest element be? Explain your answers (and provide illustrations).

(ii) Consider a set containing the elements 1,2,3,4,5,6,7. Is it possible that a min-heap built on the 7 elements is identical to a binary search tree (BST) for the 7 elements? The BST does not have to be balanced. Justify your answer.