Sample Midterm Problems

WARNING These Problems were given on a midterm I gave in 1990. At that point it was a quarter course so this was given at week 5. Chapter 3 was not included.

1) (10 pt) Let Split be the algorithm given in the text for splitting the list durring Quicksort. What list will be returned when Split is applied to the list:

5 7 1 9 2 4 8 6 0 3

2) (10pt) Suppose Heapsort is givent the array

5 7 1 9 2 4 8 6 0 3

Give the correct contents of the array after the Heap construction phase of the algorithm.

3) (10pt) Suppose we want to Shellsort the list

5 7 1 9 2 4 8 6 0 3

using increments 1,3,5.

a) What is the list after we have sorted with increment 5?

b) What is the list after we have sorted with increment 3?

4) (15pt) Find the minimal spanning tree for the graph with the following adjacency list [**NOTE**: We read A: B(1),D(2) as saying there is an edge AB of weight 1 and an edge AD of weight 2].

A: B(1),D(2)
B: A(1),C(3),D(1),G(3)
C: B(3),E(2)
D: A(2),B(1),E(2)
E: C(2),D(2),H(3)
F: B(4),E(1),G(2),H(5)
G: B(3),F(2)
H: E(3), F(5)

5) Show by example that the Shortest Path Algorithm we have discussed may not work if some edges have negative weight. Explain why?

6) (20pt) Consider the following algorithm for sorting an array L[1],. . .,L[n].

```
for i=1 to n-1 {
  min=i;
  for j=i+1 to n {
    if L[j]<L[i] then min=j;}
  temp=L[i];
  L[min]=L[i];
  L[i]=temp;}
```

Exactly how many key comparisons will this algorithm do in the worst case for a list of lenght n? Which lists will require the maximal number of key comparisons?

7) (25pt) Suppose we have a list of keys L[1],. . .,L[n] of keys, where each key may be a positive or negative number. Give an efficient in-place algorithm to rearrange the keys so that all of the negative keys precede all of the positive keys. For full credit your algorithm should examine each key only once.