Ph.D. Qualifying Exam
Design and Analysis of Algorithms and Data Structures

Computer Science Department
Indiana University

Spring 1998

This exam has two problems.

Problem I

The following sorting algorithm has been proposed:

```
STOOGE-SORT(A, i, j)
    if A[i] > A[j]
        then exchange A[i] ↔ A[j]
    if i + 1 > j
        then return
    k ← ⌊(j − i + 1)/3⌋          round down
    STOOGE-SORT(A, i, j − k)      first two-thirds
    STOOGE-SORT(A, i + k, j)      last two-thirds
    STOOGE-SORT(A, i, j − k)      first two thirds again
```

1. Argue that STOOGE-SORT$(A, 1, \mathit{length}[A])$ correctly sorts the input array $A[1..n]$, where $n = \mathit{length}[A]$.
2. Give a recurrence for the worst-case running time of STOOGE-SORT and a tight asymptotic (Θ-notation) bound on the worst-case running time.
3. Compare the worst-case running time of STOOGE-SORT with relevant classical sorting algorithms (give their names and complexities). Is STOOGE-SORT worthwhile?

Problem II

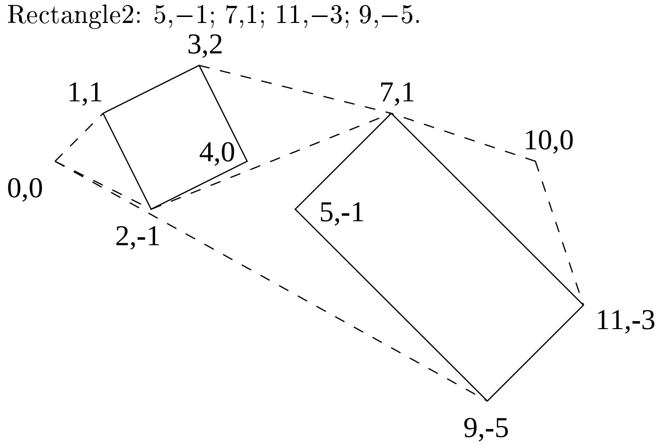
This problem is concerned with finding the shortest path (in a two dimensional space) when obstacles are present. The input consists of the starting point, the goal point, and some rectangles. Each point is specified by its X and Y coordinate. Each rectangle is specified by the points of its four corners (in clockwise order). The problem is to find the shortest path from the starting point to the goal point, where the path is not permitted to go inside of any rectangle.

Describe in English each idea that is needed to design an efficient algorithm for this problem. Be sure to be complete. For example, if you believe that you need a routine to determine whether two lines intersect, also describe how you can determine whether two lines intersect. If your discussion is complete, it should be obvious to a computer scientist how to program your algorithm.

Give a short analysis of the worst-case time bound for your algorithm.

Briefly and clearly discuss additional ideas that may improve the running time of your algorithm on typical inputs.

Example:
Source: 0,0.
Goal: 10,0.
Rectangle1: 1,1; 3,2; 4,0; 2,−1.
Rectangle2: 5,−1; 7,1; 11,−3; 9,−5.



Some paths:

$$0,0;\ 1,1;\ 3,2;\ 7,1; 10,0.\ \text{Length: } \sqrt{2}+\sqrt{5}+\sqrt{17}+\sqrt{10}=10.9$$

$$0,0;\ 2,-1;\ 7,1;\ 10,0.\ \text{Length: } \sqrt{5}+\sqrt{29}+\sqrt{10}=10.8$$

$$0,0;\ 9,-5;\ 11,-3;\ 10,0.\ \text{Length: } \sqrt{106}+\sqrt{8}+\sqrt{10}=16.3$$