ICS 164 – Spring 2008 – Midterm

Name:

Student ID:

1:

2:

3:

4:

Total:

1. (30 points) For each of the following algorithms, state the data structure(s) that are used as part of the algorithm. Data structures that may occur in these algorithms include the stack, priority queue, binary search tree, and doubly-connected edge list.

(a) Which of these data structure(s) is or are used in the Graham scan convex hull algorithm?

(b) Which of these data structure(s) is or are used in the plane sweep algorithm for reporting the intersecting pairs of line segments in a collection of line segments?

(c) Which of these data structure(s) is or are used in the incremental algorithm for constructing a line arrangement?

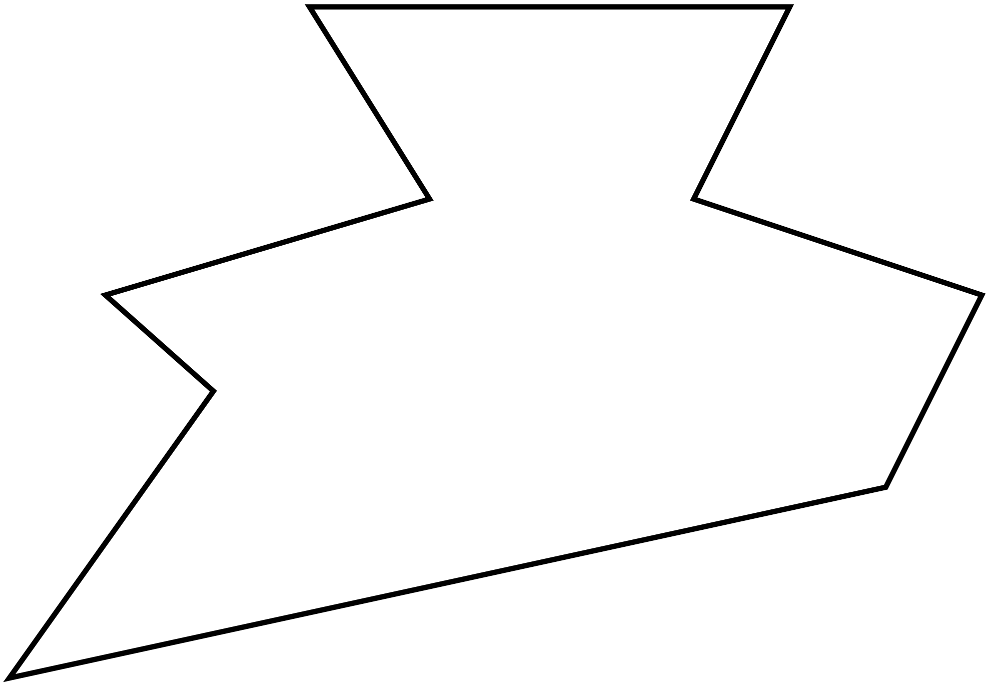
(d) Which of these data structure(s) is or are used in the randomized incremental algorithm for linear programming?

2. (30 points) A two-dimensional point p belongs to the convex hull of a collection of points q_i if and only if it can be represented as a linear combination $p = \sum a_i q_i$ in which the coefficients a_i are all in the range from zero to one and the sum of the coefficients is one.

(a) Express the problem of testing whether p belongs to the convex hull of the points q_i as a linear program. What are the variables, what are the linear constraints the variables must satisfy, and what (if any) linear objective function should be minimized?

(b) Can this linear program be solved efficiently by Seidel's randomized linear programming algorithm? Why or why not?

3. (30 points) Draw the triangulation of the polygon below that would be produced by the greedy plane-sweeping algorithm described in class.



4. (30 points) Suppose we are given a sequence of n points $P[0]$, $P[1]$, $\ldots$, $P[n-1]$, and we wish to test whether they form the clockwise vertex sequence of a convex polygon.

Consider the following algorithm:

```
def testConvex(P):
    for i = 0, 1, 2, ..., n-3:
        if points P[i], P[i+1], and P[i+2] are not in clockwise
                  order around the triangle they form:
            return False
    return True
```

Does this algorithm correctly determine whether the sequence of points forms a convex polygon? Why or why not?

You may use this page (or the back of the other pages) as scratch paper.