# CS 6463-004 Computational Geometry, Fall 06

10/5/06

# 3. Homework

Due **10/26/06** before class

1. (7.5) Parabolic Arcs – 5 points
   Give an example where the parabola defined by some site $p_i$ contributes more than one arc to the beach line. Can you give an example where it contributes a linear number of arcs?

2. Weighted Voronoi Diagrams – 10 points
   Let $P = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^2$, and let $w_i > 0$ be the *weight* of point site $p_i$, for each $i = 1, \ldots, n$. In the *additively weighted* Voronoi diagram the Voronoi cell for $p_i$ is defined as

   $$V_{add}(p_i) = \{q \in \mathbb{R}^2 \mid w_i + ||p_i - q|| < w_j + ||p_j - q|| \text{ for all } p_j \in P \setminus \{p_i\}\}$$

   In the *multiplicatively weighted* Voronoi diagram the Voronoi cell for $p_i$ is defined as

   $$V_{mult}(p_i) = \{q \in \mathbb{R}^2 \mid w_i * ||p_i - q|| < w_j * ||p_j - q|| \text{ for all } p_j \in P \setminus \{p_i\}\}$$

   Show how the bisectors look like for both kinds of weighted Voronoi diagrams, and give some examples of Voronoi diagrams for each case. You are welcome to research on the web as long as you give references.

3. (9.3) Transformation Using Edge Flips – 10 points
   Prove that any two triangulations of a planar point set can be transformed into each other by edge flips. *Hint: Show first that any two triangulations of a convex polygon can be transformed into each other by edge flips.*

4. (9.9) Worst-Case DT Runtime – 5 points
   Show that the worst-case runtime of the randomized algorithm to compute the Delaunay triangulation of a set of $n$ points in the plane is $\Omega(n^2)$. *Hint: Find a worst-case example using one of the Delaunay triangulation programs.*

5. Backwards Analysis – 10 points
   Consider the following algorithm:

```
FindMax(A,n){
  // Finds maximum in set A of n numbers
  if(n==1) return single element in A
  else{
    x = extract random element from A // in constant time; x is removed from A
    y = FindMax(A,n-1)
    if(x<=y) return y;
    else
       Compare x with all remaining elements in A and return the maximum
  }
}
```

   a) (**4 points**) Argue that this algorithm is correct, and give its worst-case runtime (The runtime is proportional to the number of comparisons made).
   b) (**6 points**) Compute the expected runtime of this algorithm. *Hint: Introduce an indicator random variable for executing the else branch in the i-th step, and use backwards analysis to simplify the analysis. Use Mount's notes (page 34) as a reference.*

FLIP OVER TO BACK PAGE ⟹

6. (7.14) Extra Credit – 10 points
**This problem is for extra credit and is not mandatory.**
Suppose we are given a subdivision of the plane into $n$ convex regions. We suspect that this subdivision is a Voronoi diagram, but we do not know the sites. Develop an algorithm that finds a set of $n$ point sites whose Voronoi diagram is exactly the given subdivision, if such a set exists.