15-451/651 Algorithms, Fall 2014
Quiz 11 Solutions

---

1. The input to the incircle test consists of how many points in 2-d?

   (d) 4

2. The incircle test is applied to the following points (0,0) (1,0) (1,1) (0,1). The result will be:

   $= 0$

3. The incircle test is applied to the following points (0,0) (1,0) (1,1) (10,0). The result will be:

   < 0.

4. What is the area of the convex combination of points (2,2) (3,3) (1,0)

   (g) 1/2

5. Consider the affine combination of two points p and q. Each option below specifies some conditions on p and q. Select those that guarantee that this affine combination contains the origin.

   p is the origin
   q is the origin
   the origin and p and q are colinear

6. In class we described a random incremental 2D LP algorithm for finding a point satisfying all n constraints while achieving the maximum value of the objective function. The expected running time of this algorithm is (best answer):

   $O(n)$.

7. The above mentioned algorithm invokes a solver for another problem. Which of these is it:

   Solving a 1D LP problem

8. The dual of a 2D LP is another 2D LP.

   False! the dimension will be the number of variables of the dual, which is equal to the number of constraints of the primal!

9. Which of the following primitives would be the most useful for computing the area of a parallelogram?

   cross product

10. You're given n points in the plane, $p_1, ..., p_n$. Define Region(p_i) to be the set of points $q \in R^2$ such that

    distance(q, p_i) $\leq \min_j$ (distance(q, p_j)).

    Select all the properties below that Region(p_i) satisfies. (You should select as many as are correct, every correct selection wins points, incorrect selections lose points.)

convex, connected, its boundaries (if it has any) are straight lines

11. In lecture we described a sweep-line algorithm for finding all the intersections among a collection of n line segments. We maintained a priority queue Q of upcoming events. We also maintained a binary search tree T of segments crossing the current sweep-line. Say the sweep-line is at position x_s. We've just finished processing x value x_s, and we're about to start with the next x value. Which of the following statements hold at this point in time?

    - (a) Q contains all of the segment endpoints whose x coordinate is greater than x_s.
    - (d) Q contains all of segment intersections among segments currently in T, and whose x coordinate is greater than x_s, and the segments are neighbors in T.

12. The best algorithm for the convex hull of n points in 2 dimensions has running time (choose the best answer)

    $O(n \log n)$, in the comparison-based model.

13. The convex hull H of a set of n points in three dimensions satisfies the following properties (choose all that are correct):

    - H has O(n) vertices
    - H has O(n) edges
    - H has O(n) faces

14. Given the vector $\mathbf{w} = (2, -1, 2, 0, 4)$ and the point $\mathbf{a} = (1, 2, 3, 1, 1)$, what is the (Euclidean) distance of $\mathbf{a}$ from the hyperplane $\mathbf{w} \cdot \mathbf{x} = 0$? Enter a numerical value like $0.2, 7.5, -3.2$ etc.

$$\frac{\mathbf{w} \cdot \mathbf{x}}{\|\mathbf{w}\|} = 10/5 = 2.$$

15. Given the following data set in 3 dimensions, run the Perceptron algorithm (as specified in class), and enter the answer below (one number per box, please). Whenever you want to pick an i for which the current $\mathbf{w}$ is incorrect, pick the data point x_i with lowest index i.

    $\mathbf{x}_1 = (-1, 1, -1), label = 1$
    $\mathbf{x}_2 = (1, 1, -1), label = -1$
    $\mathbf{x}_3 = (0, -1, 1), label = 1$

    The final solution is $(-3, -1, -1)$.

16. Which of the following statements about the perceptron algorithm are always true? (You should select as many as are correct, every correct selection wins points, incorrect selections lose points.) You may assume that taking inner products is a constanttime operation.

    - The perceptron algorithm will always terminate. *False, it may not terminate if there is no feasible solution. For instance, take two points $\mathbf{x}_1 = \mathbf{x}_2$ but $y_1 = -y_2$.*
    - The perceptron algorithm will terminate if there is a feasible solution to the given Linear Separability instance. *True, this is what we proved in class.*

- The perceptron algorithm is not a polynomial time algorithm (restricted to instances where it halts). *True: the instance in the example should give you some hint of how to prove this. One bad example would be two positive points* $\mathbf{x}_1 = (1, 0)$ *and* $\mathbf{x}_2 = (-1, 1/N)$, *with both* $y_1 = y_2 = 1$. *Note that the length of the input is* $O(\log N)$ *bits, but the perceptron algorithm takes* $\mathrm{poly}(N)$ *steps.*
- The vector it outputs is the shortest length vector that is a solution to the Linear Separability instance. *False. Many simple examples, e.g., one positive point* $\mathbf{x}_1 = (2/5, 0)$ *and* $y_1 = 1$. *The algorithm will output* $\mathbf{w} = (2.6, 0)$ *whereas the shortest vector would be* $\mathbf{w}^* = (2.5, 0)$.

17. Suppose you have a deterministic algorithm for predictions from expert advice. For any set of n experts and any sequence of days, suppose this algorithm guarantees that the number of mistakes it makes is at most $am + f(n)$ for some parameter a, where m is the number of mistakes made by the best expert on that sequence, and $f(n)$ is some function solely of the number of experts. What is the smallest is it possible to make a ? Note: we are not asking just for the smallest described in the lecture notes, we're talking about how small you can make using any deterministic algorithm. You should be able to prove that no deterministic algorithm can do better than your claimed value of a.

    By changing the "reduction" factor from $1/2$ to $1 - \epsilon$, and choosing the weighted majority deterministically, you can get that $M \leq 2(1 + \epsilon) + O(\frac{\log n}{\epsilon})$. So you can get as close to 2 as you want.

    On the other hand, you cannot get $a < 2$ with a deterministic algorithm, even when there are only 2 experts. Consider the setting where there are 2 experts, the first E_1 who always says Y and the other E_2 who always says N. Every day i your algorithm gets two predictions (Y from E_1 and N from E_2), and makes a prediction P_i. *Since it is deterministic algorithm this prediction is completely determined by the past outcomes.* I.e., $P_i = f(O_1, O_2, \ldots, O_{i-1})$. Suppose the outcome O_i is the opposite of P_i.

    So you are wrong every single time. In T steps, $M = T$. But what about the best expert? Look at the majority of $O_1, O_2, \ldots, O_T$. If this is Y then E_1 makes at most $T/2$ mistakes, else E_2 makes at most $T/2$ mistakes. $M \geq 2m$.

18. Suppose you use the following algorithm for learning from expert advice: Suppose you have n experts. You predict what the majority of your current set of experts say. And after every round you drop the experts that were incorrect. When all experts have been dropped, you bring all experts back, and continue. If the best expert makes m mistakes, and you make M mistakes, what is the best bound you can give on M?

    $M = O(m \log n)$.