Algorithms
Problem Set 3
University of Virginia

Gabriel Robins

Please make all algorithms as efficient as you can, and state their time and space complexities.

1-46. Solve the following problems from the [Cormen, Third Edition, 2009] algorithms textbook:

p. 456: 17.1-1, 17.1-2, 17.1-3
p. 458-459: 17.2-1, 17.2-3
p. 462-463: 17.3-2, 17.3-4, 17.3-6, 17.3-7
p. 471: 17.4-1
p. 473: 17-2
p. 593: 22.1-5, 22.1-6, 22.1-7, 22.1-8,
p. 602: 22.2-4, 22.2-7, 22.2-8, 22.2-9,
p. 612: 22.3-12, 22.3-13
p. 614-615: 22.4-2, 22.4-3, 22.4-4, 22.4-5
p. 621: 22.5-7
p. 623: 22-3, 22-4
p. 629-630: 23.1-1, 23.1-5, 23.1-6, 23.1-7, 23.1-11
p. 637-640: 23.2-4, 23.3-5, 23.2-6, 23.2-7, 23-1, 23-3,
p. 655: 24.1-6
p. 658: 24.2-4
p. 663-664: 24.3-6, 24.3-8, 24.3-9
p. 678-679: 24-2, 24-3,

47. Give an **optimal** algorithm to determine the diameter of a weighted tree (i.e., longest path between any two nodes). Prove the optimality of your algorithm. What is your time complexity?

48. Give an algorithm to determine whether any two of N segments on a line intersect.

49. Give an algorithm to find the N-way intersection of N rectangles with sides parallel to the axis.

50. Give an algorithm to compute the diameter of a given pointset (i.e., the distance between the farthest pair of points).

51. Give a linear-time algorithm to compute the area of a given convex polygon.

52. Give a linear-time algorithm to compute the area of a given non-convex polygon.

53. What is the worst-case cost of a traveling salesman tour over n points located arbitrarily in the unit square? Prove it. (Express the cost asymptotically as O(f(n)) for some function f.)

54. What is the expected (average) cost of a minimum spanning tree over n points uniformly distributed in the unit square? (Express the cost asymptotically as O(f(n)) for some function f.)

55. Given N **red** points and N **blue** points arbitrarily placed in the plane, we seek to match these 2N points into N pairs, so that no two of the line segments defined by these pairs intersect. (Assume that no three of the 2N distinct points are collinear.) Prove or disprove: there always exists such a non- self-intersecting matching for any arrangement of N red and N blue points arbitrarily located in the plane. Devise an efficient algorithm that given an arbitrary set of 2N red & blue points in the plane, finds such a matching, when one exists. Generalize all this to dimension D.