MA/CSSE 473 Day 15

Answer Question 1

Answers to your questions

Divide and Conquer

Closest Points

DIVIDE AND CONQUER

Divide-and-conquer algorithms

- Definition
- Examples seen prior to this course or so far in this course

Today: Closest Points, Convex Hull

DIVIDE AND CONQUER ALGORITHMS

Divide-and-conquer algorithms

- Definition
- Examples seen prior to this course or so far in this course

Q1

Closest Points problem

- Given a set, S, of N points in the xy-plane, find the minimum distance between two points in S.
- Running time for brute force algorithm?
- Next we examine a divide-and-conquer approach.

Closest Points "divide" phase

- S is a set of N points in the xy-plane
- For simplicity, we assume $N = 2^k$ for some k.
- Sort the points by x-coordinate
  - If two points have the same x-coordinate, order them by y-coordinate
  - If we use merge sort, the worst case is $\Theta(N \log N)$
- Let c be the median x-value of the points
- Let S_1 be $\{(x, y): x \le c\}$, and S_2 be $\{(x, y): x \ge c\}$
  - adjust so we get exactly N/2 points in each subset

Q2

Closest Points "conquer" phase

- Assume that the points of S are sorted by x-coordinate, then by y-coordinate if x's are equal
- Let d_1 be the minimum distance between two points in S_1 (the set of "left half" points)
- Let d_2 be the minimum distance between two points in S_2 (the set of "right half" points)
- Let $d = \min(d_1, d_2)$. Is d the minimum distance for S?
- What else do we have to consider? Q3
- Suppose we needed to compare every point in S_1 to every point in S_2. What would the running time be? Q4
- How can we avoid doing so many comparisons?

Reference: The Master Theorem

- The Master Theorem for Divide and Conquer recurrence relations:
- Consider the recurrence $T(n) = aT(n/b) + f(n)$, $T(1)=c$, where $f(n) = \Theta(n^k)$ and $k \geq 0$,
- The solution is
  - $\Theta(n^k)$ if $a < b^k$
  - $\Theta(n^k \log n)$ if $a = b^k$
  - $\Theta(n^{\log_b a})$ if $a > b^k$

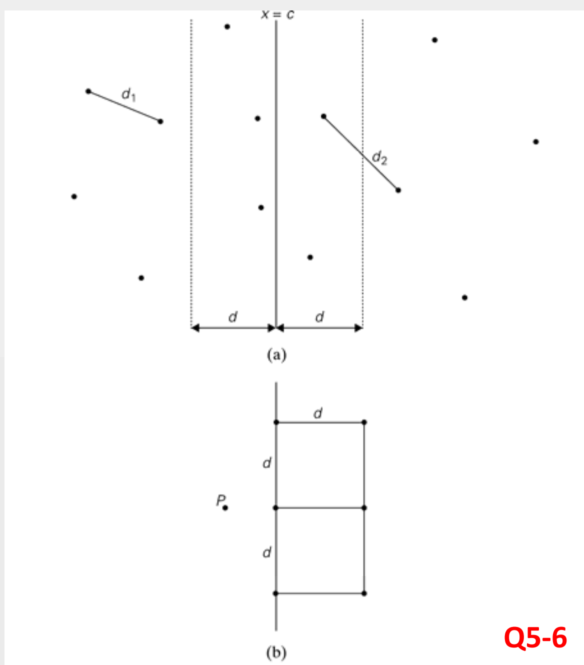
For details, see Levitin pages 483-485 or Weiss section 7.5.3.

Grimaldi's Theorem 10.1 is a special case of the Master Theorem.

We will use this theorem often. You should review its proof soon (Weiss's proof is a bit easier than Levitin's).

After recursive calls on S_1 and S_2

$d = \min(d_1, d_2)$.



Q5-6

FIGURE 4.7 (a) Idea of the divide-and-conquer algorithm for the closest-pair problem. (b) The six points that may need to be examined for point P.