Closest pair of points in R^2: algorithm and run time analysis

CSC373: Algorithm Design, Analysis, and Complexity

Sara Rahmati (in collaboration with Allan Borodin)

University of Toronto

Winter 2020

Closest pair of points (from Kevin Wayne)

```
Closest-Pair(p₁, …, pₙ) {
   Compute separation line L such that half the points
   are on one side and half on the other side.

   δ₁ = Closest-Pair(left half)
   δ₂ = Closest-Pair(right half)
   δ  = min(δ₁, δ₂)

   Delete all points further than δ from separation line L

   Sort remaining points by y-coordinate.

   Scan points in y-order and compare distance between
   each point and next 7 neighbors. If any of these
   distances is less than δ, update δ.

   return δ.
}
```

Compute separation line	$O(n \log n)$
Recursive calls	$2T(n / 2)$
Delete points	$O(n)$
Sort	$O(n \log n)$
Scan	$O(n)$

Closest pair of points

Running time.

$$T(n) \leq 2T(n/2) + O(n \log n) \quad \Rightarrow \quad T(n) = O(n \log^2 n)$$

Master theorem does not directly apply. Need to unroll the recurrence. (Cf. CLRS Exercise 4.4.2).

Q. Can we achieve O(n log n)?

A. Yes. Presort points in two lists, one sorted by x, other by y.

- Each recursive call accepts two sorted lists: one sorted by y the other by x.
- Finding a splitting in x is O(1).
- Computing the y-sorted list in the strip is O(n).

$$T(n) \leq 2T(n/2) + O(n) \quad \Rightarrow \quad T(n) = O(n \log n)$$

Time complexity analysis

- Goal: prove $T(n) \le 2T(n/2) + O(n \log n); T(2) = 1 \implies T(n) \le n \log^2 n$

Base case

- Proof: For simplicity, assume n = 2^m for m a non-negative integer:
- Transform variable n to m: $n = 2^m \implies \log_2 n = m, \implies \log_2(n/2) = m - 1$

$$
\begin{aligned}
T'(m) &\le 2T'(m-1) + O(2^m \times m) \\
&\le 2^2 T'(m-2) + O(2^m \times ((m-1) + m)) \\
&\le 2^3 T'(m-3) + O(2^m \times ((m-2) + (m-1) + m)) \\
&\cdots \\
&\le 2^m T'(0) + O(2^m \times (1 + \ldots + (m-2) + (m-1) + m)) \\
&= 2^m T'(0) + O(2^m \times (m(m+1)/2), \\
T'(1) &= 1
\end{aligned}
$$

Base case (Since T(2) = 1))

- Inverse transform variable m to n: $T(n) \le n + O(n(\log n \log(2n)/2) = O(n \log^2 n)$