

MAT2540, Classwork12, Spring2026

8.3 Divide-and-Conquer Algorithms and Recurrence Relations (Conti.)

6. Theorem of size estimation of the function f (p. 556).

Let f be an increasing function that satisfies the recurrence relation

$$f(n) = \alpha f\left(\frac{n}{\beta}\right) + c$$

whenever n is divisible by an integer $\beta > 1$, where $\alpha \geq 1$, and c is a positive real number. Then

$$f(n) \text{ is } \begin{cases} O(n^{\log_{\beta} \alpha}) & \text{if } \alpha > 1, \\ O(\log n) & \text{if } \alpha = 1. \end{cases}$$

$$\begin{aligned} f(n) &= \alpha^k f(1) + \sum_{j=0}^{k-1} \alpha^j c \\ &= \alpha^k f(1) + \sum_{j=0}^{k-1} \alpha^j \cdot c \\ &= \alpha^k f(1) + c \cdot \sum_{j=0}^{k-1} \alpha^j \end{aligned}$$

Furthermore, when $n = \beta^k$ and $\alpha \neq 1$, where k is a positive integer, we have

$$k = \log_{\beta} n \quad f(n) = C_1 n^{\log_{\beta} \alpha} + C_2,$$

where $C_1 = f(1) + \frac{c}{\alpha-1}$ and $C_2 = -\frac{c}{\alpha-1}$.

Observation: Using the result from 5: $f(n) = \alpha^k f(1) + \sum_{j=0}^{k-1} \alpha^j c = \alpha^k f(1) + c \sum_{j=0}^{k-1} \alpha^j$. We have $k = \log_{\beta}(n)$

1) when $\alpha = 1$, $f(n) = 1 \cdot f(1) + c \cdot k = f(1) + c \cdot \log_{\beta} n$ implies $f(n)$ is $O(\log_{\beta} n)$

2) when $\alpha > 1$, $f(n) = \alpha^k f(1) + c \sum_{j=0}^{k-1} \alpha^j = \alpha^k f(1) + c \cdot \frac{\alpha^k - 1}{\alpha - 1}$ implies $f(n)$ is $O(n^{\log_{\beta} \alpha})$

$$= \alpha^k f(1) + c \frac{\alpha^k - 1}{\alpha - 1} = \alpha^k \left(f(1) + \frac{c}{\alpha - 1} \right) - \frac{c}{\alpha - 1}$$

7. Let $f(n) = 5f\left(\frac{n}{2}\right) + 3$ and $f(1) = 7$. Find $f(2^k)$, where k is a positive integer. Also estimate $f(n)$ if f is an increasing function.

$\alpha = 5$
 $\beta = 2$
 $c = 3$

$$\alpha > 1, f(2^k) = 5^k f(1) + 3 \cdot \frac{5^k - 1}{4} = 5^k \cdot 7 + \frac{3}{4} \cdot 5^k - \frac{3}{4} = 5^k \left(7 + \frac{3}{4} \right) - \frac{3}{4}$$

Also f is increasing, $f(n)$ is $O(n^{\log_2 5})$

8. Give a big- O estimate for the number of comparisons used by a binary search.

$$f(n) = 1 \cdot f\left(\frac{n}{2}\right) + 2 \quad c=2$$

let $n = 2^k$, $\alpha = 1$, $\beta = 2 \rightarrow k = \log_2 n$

$$f(n) = f(1) + 2 \cdot k = f(1) + 2 \cdot \log_2(n) \Rightarrow f(n) \text{ is } O(\log_2(n))$$

$$\log_{\beta} n = \frac{\log n}{\log \beta} = \frac{1}{\log 2} \cdot \log n$$

9. Master Theorem (p. 558).

Let f be an increasing function that satisfies the recurrence relation

$$f(n) = \alpha f\left(\frac{n}{\beta}\right) + cn^d$$

whenever n is divisible by an integer $\beta > 1$, where $\alpha \geq 1$, and $c > 0$ and $d \geq 0$. Then

$$f(n) \text{ is } \begin{cases} O(n^d) & \text{if } \alpha < \beta^d \\ O(n^d \log n) & \text{if } \alpha = \beta^d, \\ O(n^{\log_{\beta} \alpha}) & \text{if } \alpha > \beta^d. \end{cases}$$

10. Give a big- O estimate for the number of comparisons used by the merge sort to sort a list of n elements.

11. The Closest-Pair Problem (a divide-and-conquer algorithm from computational geometry)

Consider the problem of determining the closest pair of points in a set of n points $(x_1, y_1), \dots, (x_n, y_n)$ in the plane, where the distance between two points (x_i, y_i) , and (x_j, y_j) is $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. How can this closest pair of points be found in an efficient way?

Direct way without divide-and-conquer: _____.

Efficient way with divide-and-conquer:

1) Assume that $n = 2^k$. When $n = 2$, the distance between these 2 points is _____.

2) Using merge sort twice: once to sort the points in order of increasing x coordinates, and once to sort the points in order of increasing y coordinates. Each of these sorts requires $O(n \log n)$.

3) The recursive part of the algorithm divides the problem into two subproblems.

a) Split the n points into left/right region:

Find a vertical line ℓ dividing the n points into two parts using the x -coordinates sorting list.

b) Find the minimum distance in left/right region recursively.

The total process requires _____ comparisons.

There are three cases to find the minimum distance:

d_L (The minimum distance in the _____), d_R (The minimum distance in the _____), and d_{RL} (the distance of a point _____ and a point _____)

4) Find d_L, d_R by step 3). Let $d = \min(d_L, d_R)$.

5) To find the if any d_{RL} is smaller than d , we do the following:

a) Only consider the points around line ℓ within _____ since _____

b) Check distances starting from the point with smallest y -coordinate in this strip.

c) Let the point with smallest y -coordinate be p_i . We only need to check the distances between p_i and points in the set that lie within _____ and with p_i on its base and with vertical sides at distance d from ℓ since _____

d) There are at most _____ other points within this $d \times 2d$ rectangle.

e) Once the distance checking of p_i is done, we move to p_{i+1} for $i < n$.

Then the total comparisons in step 5) is at most _____.

6) Finally, we have the increasing recurrence relation _____.

