Study Group – Divide and Conquer

Name: ______________________ Wisc id: ______________________

Solving recurrences

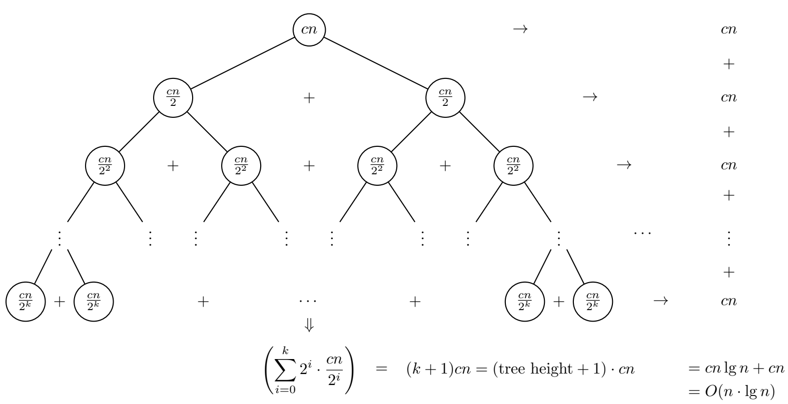
$$T(n) \le 2 \cdot T\left(\frac{n}{2}\right) + cn; T(1) \le c$$

Unrolling / unwinding:

$$\begin{aligned} T(n) &\le 2T\left(\frac{n}{2}\right) + cn \\ &\le 2\left(2T\left(\frac{n}{4}\right) + c\frac{n}{2}\right) + cn \\ &\le 2\left(2\left(2T\left(\frac{n}{2^3}\right) + c\frac{n}{2^2}\right) + c\frac{n}{2}\right) + cn \\ &\vdots \\ &\le 2^k T\left(\frac{n}{2^k}\right) + kcn \\ &= nT(1) + cn\log(n), \text{ using (1)} \\ &= cn + cn\log n \\ &= O(n\log(n)) \end{aligned}$$

$$\begin{aligned} 1 &= \frac{n}{2^k} \\ \iff 2^k &= n \\ \iff k &= \log_2(n) \end{aligned} \tag{1}$$

Recursion Tree:



Master Theorem

If $T(n)$ is defined by a standard recurrence, of the form

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + O(n^d)$$

where $a \geq 1, b > 1, d \geq 0$.

The parameter a represents the number of recursive calls.

The parameter b represents the factor by which the input size shrinks in recursive calls.

The parameter d represents the exponent of the work done outside of the recursive calls.

Then, by the Master Theorem:

$$T(n) = \begin{cases} O(n^d \log n), & a = b^d \\ O(n^d), & a < b^d \\ O\left(n^{\log_b a}\right), & a > b^d \end{cases}$$

Problems

1. Given a sorted array of distinct integers $A[1, \ldots, n]$, you want to find out whether there is an index i for which $A[i] = i$. Design an $O(\log n)$ time algorithm to find an index i if it exists.

2. Given an $n \times n$ matrix A and a power k. Give a divide-and-conquer algorithm that computes A^k. For example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 6 & 3 \\ 4 & 9 & 1 \end{bmatrix}^2 = \begin{bmatrix} 17 & 41 & 12 \\ 26 & 67 & 27 \\ 26 & 71 & 40 \end{bmatrix}, \qquad \begin{bmatrix} 1 & 2 & 3 \\ 2 & 6 & 3 \\ 4 & 9 & 1 \end{bmatrix}^6 = \begin{bmatrix} 186601 & 483340 & 207636 \\ 326560 & 846221 & 364416 \\ 381368 & 988728 & 426997 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}, \text{where } c_{ij} = \sum_{k=1}^{n} a_{ik} \cdot b_{kj}$$

Hint: Matrix multiplication is associative, i.e., $(A\,B)\,C = A\,(B\,C)$