

Algorithm Design and Analysis (Fall 2022)

Assignment 1

Deadline: Mar 19, 2022

1. (20 Points) Prove the following generalization of the master theorem. Given constants $a \geq 1, b > 1, d \geq 0$, and $w \geq 0$, if $T(n) = 1$ for $n < b$ and $T(n) = aT(n/b) + n^d \log^w n$, we have

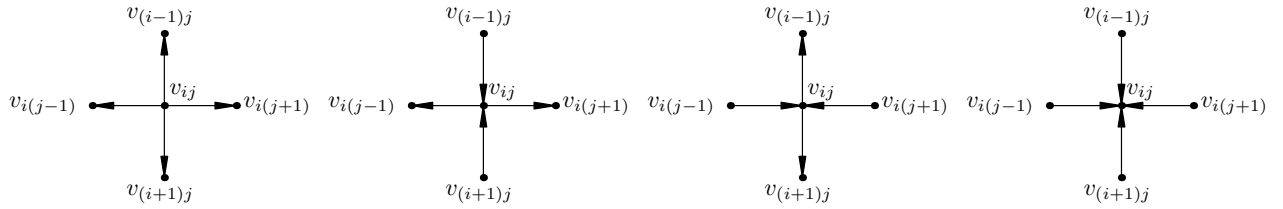
$$T(n) = \begin{cases} O(n^d \log^w n) & \text{if } a < b^d \\ O(n^{\log_b a}) & \text{if } a > b^d \\ O(n^d \log^{w+1} n) & \text{if } a = b^d \end{cases}.$$

2. (20 points) Recall the median-of-the-medians algorithm we learned in the lecture. It groups the numbers by 5. What happens if we group them by 3, 7, 9...? Please analyze those different choices and discuss which one is the best. Note that in this problem, we may drop the big- O notation and discuss the constants.
3. (20 points) Given n integers, where n is even. Can you find both the maximum and minimum within $3n/2 - 2$ comparisons? Note that we also do not use big- O notations here.
4. (20 points) Recall the Word RAM model we learn in the lecture. Let w be the number of bits in a machine word. We know that those basic operations on each word (like addition, right shift, and/or, ...) can be viewed as unit operations. We will design algorithms to compute the transpose A^T of a binary matrix $A \in \{0, 1\}^{h \times h}$.
- (a) (5 points) If we want to store the matrix in one word, how large h can be? How to store it in one word?
- (b) (15 points) Consider an one-word size matrix A . Design an $O(\log w)$ algorithm to compute its transpose A^T .

5. (20 points) Let $G = (V, E)$ be a $n \times n$ grid which is an undirected graph with n^2 vertices labeled $V = \{v_{ij}\}_{i=0,1,\dots,n-1;j=0,1,\dots,n-1}$. Vertex v_{ij} is adjacent to $v_{(i-1)j}$ (unless $i = 0$), $v_{(i+1)j}$ (unless $i = n - 1$), $v_{i(j-1)}$ (unless $j = 0$), and $v_{i(j+1)}$ (unless $j = n - 1$).

Let H be a directed graph sharing the same vertex set V as G . The edge set of H is the same as the edge set of G , except that a direction of each edge in G is specified for the corresponding edge in H .

- (a) (10 points) Suppose the directions of the edges in H satisfy the followings. The directions of the edges between v_{ij} and the “left” and the “right” neighbors of v_{ij} are the same, i.e., either both edges point towards v_{ij} or both edges point outwards v_{ij} , and the directions of the edges between v_{ij} and the “upper” and the “lower” neighbors of v_{ij} are the same. Specifically, the four edges incident to v_{ij} (where $0 < i < n - 1$ and $0 < j < n - 1$) must be in one of the following four configurations.



Design an algorithm that finds *one vertex with out-degree 0 or one directed cycle*. Your algorithm must run in $O(\log n)$ time.

- (b) (10 points) Suppose the directions of the edges in H are defined as follows. There is a function (given as input) $f : V \rightarrow \mathbb{Z}$ that assigns an integer *value* to each vertex, and assume $f(u) \neq f(v)$ for any $u \neq v \in V$. The direction of the edge between the two vertices u, v is from the vertex with a higher value to the vertex with a lower value.

Design an algorithm that finds *one vertex with out-degree 0 or one directed cycle*. Your algorithm must run in $O(n)$ time.

- (c) (Bonus 5 points) If no further assumptions were made for directions of the edges in H , prove that any algorithm that finds *one vertex with out-degree 0 or one directed cycle* requires $\Omega(n^2)$ time.

6. How long does it take you to finish the assignment (including thinking and discussion)? Give a score (1,2,3,4,5) to the difficulty. Do you have any collaborators? Please write down their names here.