Fundamental Algorithms, Section 003
Homework 3, Fall 22. Due date: Wednesday, September 28, 4:55pm on Gradescope

1. Solve the following recurrence equations exactly. Don't try to use the Master method from the textbook; rather, use the recursion tree method shown in class.
a. Suppose that $n = 2^{2^k}$ for some integer $k \geq 0$.

$$T(n) = 4\sqrt{n}T(\sqrt{n}) + n \quad n > 2$$
$$T(2) = 1$$

b. Suppose that $n = 2^k$ for some integer $k \geq 0$.

$$T(n) = T(n/2) + \log n \quad n > 1$$
$$T(1) = 1$$

Remember to check your solutions match the base case and the next larger value of n.

2. Let $A[1:n]$ be an array of integer values, which we view as points on a line. The task is to determine how many pairs of points are distance d or less apart; d is an input parameter. Give an $O(n \log n)$ time algorithm to solve this problem. Remember to analyze the running time of your algorithm.

3. Suppose you are given an array $A[d:u]$ of distinct integers, and an integer r in the range $[1, u-d+1]$. Suppose further that you have an algorithm ApproxSelect(A, d, u, r, i, k) which reorders the array A and returns two locations i and k in the array with the following properties:

- Items of value smaller that $A[i]$ lie to the left of $A[i]$, items larger than $A[k]$ lie to its right, and the remaining items, those with values strictly between $A[i]$ and $A[k]$, lie between $A[i]$ and $A[k]$:
  - $A[h] < A[i]$ for $d \leq h < i$.
  - $A[i] < A[j] < A[k]$ for $i < j < k$.
  - $A[k] < A[\ell]$ for $k < \ell \leq u$.
- the r-th smallest item lies between $A[i]$ and $A[k]$: $i - d + 1 \leq r \leq k - d + 1$.
- If $n = u - d + 1$ is the number of items in the array, then the number of items in $A[i+1:k-1]$, the number of items with value strictly between $A[i]$ and $A[k]$, is at most $n/2$.
- ApproxSelect runs in worst case $O(n)$ time.

Then, using ApproxSelect as a subroutine, give an algorithm Find(A, d, u, r) to find the r-th smallest item in an array $A[d:u]$ running in worst case time $O(n)$.

You should describe your algorithm and analyze its running time.
Hint: Draw a picture to show what ApproxSelect does.

4. Multiplying a vector by the Vandermonde matrix. The $n \times n$ Vandermonde matrix V_n is given by:

$$V_n(x) = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & x & x^2 & \dots & x^{n-1} \\ 1 & x^2 & x^4 & \dots & x^{2(n-1)} \\ & & \dots & & \\ 1 & x^{n-1} & x^{2(n-1)} & \dots & x^{(n-1)^2} \end{bmatrix}$$

Suppose $x^n = 1$ (i.e. x is an nth root of unity, which exist for arithmetic mod n when n is a prime number, and for complex numbers).

Let $\boldsymbol{a} = (a_0, a_1, \dots, a_{n-1})$ be a length n vector. We want to compute the matrix-vector product $\boldsymbol{b} = V_n(x)\boldsymbol{a}$. Remember, this product is a length n vector $\boldsymbol{b} = (b_0, b_1, \dots, b_{n-1})$, with the b_i being the dot product of the i-th row of V_n (numbering the rows as $0, 1, \dots, n-1$) with vector $\boldsymbol{a}$; in more detail, $b_i = \sum_{j=0}^{n-1} x^{ij} a_j$.

You are to obtain an $O(n \log n)$ divide-and-conquer algorithm to compute $\boldsymbol{b}$. For simplicity, assume that $n = 2^k$ for some integer k.

We define $\boldsymbol{a}^e = (a_0^e, a_1^e, \dots, a_{n/2-1}^e) = (a_0, a_2, \dots, a_{n-2})$ and $\boldsymbol{a}^o = (a_0^o, a_1^o, \dots, a_{n/2-1}^o) = (a_1, a_3, \dots, a_{n-1})$.

Let $0 \le i < n/2$. Show that $b_i = [V_{n/2}(x^2)\boldsymbol{a}^e]_i + x^i[V_{n/2}(x^2)\boldsymbol{a}^o]_i$, and $b_{n/2+i} = [V_{n/2}(x^2)\boldsymbol{a}^e]_i + x^{n/2+i}[V_{n/2}(x^2)\boldsymbol{a}^o]_i$. The index i refers to the ith entry in a vector.

Analyze the runtime of your algorithm. You may assume that all arithmetic operations take constant time.

Comment. This is the Fast Fourier Transform. It is widely used in signal processing. Don't try and read up on this; this is liable to be confusing as there are many ways of presenting this topic.

Challenge problem. Do not submit.

Problem 3.4 in the textbook.