CS 3510 A Homework 2

John Schmidt
Fall 2016

Question 1

We studied in class a linear time deterministic algorithm to find the k-th smallest element of a list of n numbers A[1] through A[n]. The following is a high-level description of a generalization of this algorithm, in which the algorithm covered in class corresponds to the case d = 2 (Here d $\geq$ 1 is an integer constant):

- Step 1: Divide the list into $\frac{n}{2d+1}$ groups of (2d + 1) elements each. This takes O(n) time.
- Step 2: Sort each group and find their medians. Let S be the sequence of medians of these groups. This takes O(n) time.
- Step 3: Recursively find the median of S. Let p be the median of S. (The size of S is $\frac{n}{2d+1}$.)
- Step 4: Partition the list A around this pivot into three sub-lists: $A_{<}p$, $A_{>}p$, and $A_{=}p$. This takes O(n) time.
- Step 5: Recurse appropriately on one of these three sub-lists.

(a) (5 points) Show that the size of each of $A_{<}p$ and $A_{>}p$ in Step 4 above is at most:

$$n - (d+1) \times \frac{n}{2(2d+1)} \tag{1}$$

(b) (5 points) What is the recurrence for the running time of this algorithm in terms of n and d? Explain the recurrence.

(c) What is the solution to this recurrence for:
i. (3 points) d = 1?
ii. (2 point) d = 3?
Explain your answers.

Solution:
Correctness Argument:
Algorithm:
Proof:
Analysis:

Question 2

(15 points) Describe an algorithm that, given an n-digit decimal integer a, outputs the square of that integer in O(n log n) time. Argue that your algorithm runs in O(n log n) time and that it is correct.

Solution:

Correctness Argument:

Algorithm:

Proof: Analysis:

Question 3
Consider the butterfly network discussed in class.
Let N = 8. The network has four columns of 8 nodes each. Let the columns be C_0, C_1, C_2, C_3. Label the nodes in column C_i as C_{i0}, $C_{i1}, \ldots$, C_{i7}. For $0 \le i \le 2$, $0 \le j \le 7$, from each node C_{ij} in column i there are two edges (a left edge and a right edge) to nodes in column i + 1. Label the left edge as 0 and the right edge as 1. Denote this network by $\beta 8$.

A path in this network is such that: (a) it has three edges, and (b) it goes from a node in column 0 to a node in column 1 to a node in column 2 to a node in column 3.

Let (a0, a1, ..., a7) be a permutation of (0, 1, 2, 3, 4, 5, 6, 7). We say that the network realizes the permutation (a0, a1, ..., a7) if there are eight vertex-disjoint paths connecting $C_{0,j}$ to $C_{3,aj}$ for $0 \le j \le 7$.

Example: The network realizes the permutation (3, 6, 1, 4, 7, 2, 5, 0) with the following eight vertex-disjoint paths:

$C_{0,0} \rightarrow C_{1,1} \rightarrow C_{2,3} \rightarrow C_{3,3}$
$C_{0,1} \rightarrow C_{1,0} \rightarrow C_{2,2} \rightarrow C_{3,6}$
$C_{0,2} \rightarrow C_{1,3} \rightarrow C_{2,1} \rightarrow C_{3,1}$
$C_{0,3} \rightarrow C_{1,2} \rightarrow C_{2,0} \rightarrow C_{3,4}$
$C_{0,4} \rightarrow C_{1,5} \rightarrow C_{2,7} \rightarrow C_{3,7}$
$C_{0,5} \rightarrow C_{1,4} \rightarrow C_{2,6} \rightarrow C_{3,2}$
$C_{0,6} \rightarrow C_{1,7} \rightarrow C_{2,5} \rightarrow C_{3,5}$
$C_{0,7} \rightarrow C_{1,6} \rightarrow C_{2,4} \rightarrow C_{3,0}$

(a) (10 points) Identify eight vertex-disjoint paths that connect C_{0j} to $C_{3,aj}$ for $0 \le j \le 7$, where

$$a_j = (\frac{5j^2 + 5j + 10}{2})\ mod\ 8. \tag{2}$$

(b) (5 points) Show, with a counter-example, that this network cannot realize all permutations of (0, 1, 2, 3, 4, 5, 6, 7).

Solution:
Correctness Argument:
Algorithm:
Proof:
Analysis:

Question 4
Let A(x) = $a_0 + a_1x + \ldots + a_{N-1}x^{N-1}$ be a polynomial with N coefficients. Assume N is a power of 3.
Divide A(x) as the sum of three polynomials as shown below:

$$A(x) = p_1(x^3) + xq_1(x^3) + x^2r_1(x^3), where \tag{3}$$

$p_1(x) = a_0 + a_3x + \ldots + a_{N-3}x^{N/3-1}$
$q_1(x) = a_1 + a_4x + \ldots + a_{N-2}x^{N/3-1}$, and
$r_1(x) = a_2 + a_5x + \ldots + a_{N-1}x^{N/3-1}$.

Let be a principal N-th root of unity. Recall that an element is a principal N-th root of unity if it satisfies the following two properties:

- $\omega^N = 1$.
- $\omega^j \neq 1$ for $1 \leq$ j $\leq$ N - 1.

Let $\omega 1$ be the inverse of . That is, $\omega^{-1} \times \omega = 1$.
(a) Assume that the following two claims are true:
Claim 1: For all 0 $\leq$ j $\leq$ N/3 - 1,
$(\omega^j)^3 = (\omega^{j+N/3})^3 = (\omega^{j+2N/3})^3$.

Corollary: For all 0 $\leq$ j $\leq$ N/3 - 1, A(ω^j) = A($\omega^{j+N/3}$) = A($\omega^{j+2N/3}$).

Claim 2: Let ω be a principal N-th root of unity. Then, ω^3 is a principal N/3-rd root of unity.

(b) (10 points) What does the following algorithm SPOLY compute? Justify your answer.
SPOLY(A, N) $\rightarrow$ R

i EVAL (A, N, ω) $\rightarrow$ U

ii For 0 $\leq$ j $\leq$ N - 1 do:

  (a) V [j] = U[j] $\times$ U[j]

iii EVAL(V, N, ω^{-1}) $\rightarrow$ R

iv For 0 $\leq$ j $\leq$ N - 1 do:

  (a) R[j] = $\frac{R[j]}{N}$

The procedure EVAL used by the algorithm above is:
EVAL(A, N, ω) $\rightarrow$ U

i If N = 1 Then Return(A[0]).

ii A1 $\leftarrow$ EV AL(p_1, $\frac{N}{3}$, ω^3)

iii A2 $\leftarrow$ EV AL($q_1, \frac{N}{3}, \omega^3$)

iv A3 $\leftarrow$ EV AL($r_1, \frac{N}{3}, \omega^3$)

v For $0 \leq j \leq N - 1$ Do:

(a) $U(j) = A1(j) + \omega^j A2(j) + \omega^{2j} A3(j)$

(c) (5 points) Write a recurrence for the run-time T(n) for the algorithm SPOLY and solve it. Explain your recurrence.

Solution:
Correctness Argument:
Algorithm:
Proof:
Analysis: