Topics in Algorithms - Exercises

November 27, 2003

Exercise 1 Implement an additional procedure for binomial heaps: $Binomial\_Heap\_Split(H, r)$ that receives as input a heap H with n elements and outputs two heaps H_1 and H_2 such that H_1 has r elements and H_2 has $n - r$ elements. Prove correctness and analyze complexity.

Exercise 2 Implement an additional procedure for Fibonacci heaps: $Fib\_Change\_Key(H, x, k)$ that receives as input a Fibonacci heap H with n nodes, an element $x \in H$ and a new key value k for x. Analyze the amortized running time of your implementation for the cases in which k is greater than, less than, and equal to $key[x]$.

Exercise 3 A *Voronoi diagram* is a data structure built on a given set of n points S in the plane. A Voronoi diagram allows, given a query point q, to find the point in S closest to q in time $O(\log n)$. Given n points in the plane, building a Voronoi diagram takes time $O(n \log n)$.

Suggest a data structure (which uses Voronoi diagrams as a black box) built on a set S of points in the plane with the following features:

1. **Insert**: A point can be added to the data structure in amortized time complexity of $O(\log^2 n)$.

2. **Find**: Given a query point q, the closest point to q can be found in amortized and worst case time complexity of $O(\log^2 n)$.

Remark: One can also solve the problem when the complexities are $O\left(\frac{\log^3 n}{\log \log n}\right)$ and $O\left(\frac{\log^2 n}{\log \log n}\right)$.

Exercise 4 Given a permutation $\pi = \pi(1) \dots \pi(n)$, a π-network is a comparison network of (input) size n with the following property: For every input $\langle a_1 \dots a_n \rangle$, the output of the network $\langle b_1 \dots b_n \rangle$ satisfies the statement "b_i is the $\pi(i)$'th smallest element among the input elements". For example if $n = 4$ and $\pi = 2, 3, 4, 1$, on every input $a_1 \dots a_4$, sorting the output of the network $b_1 \dots b_4$ will yield the order b_4, b_1, b_2, b_3. For $\pi = 1, 2, \dots, n$, a π-network is a sorting network.

Let N be a sorting network, which has m comparators. Assume that N has two kinds of comparators - the regular kind in which the minimum is obtained at the top and the opposite kind in which the minimum is obtained at the bottom. Suggest an algorithm that transforms N into a sorting network N' with m regular comparators.

Outline of a possible solution: First for a π-network N with m mixed comparators, suggest a natural way to transform N into a π'-network N' with m regular comparators. Conclude that N' is a sorting network.

Exercise 5 Prove that a network will sort the input $\langle n, n-1, n-2, \dots, 1 \rangle$ iff for each i it sorts $1^i 0^{n-i}$.

Exercise 6 Let $\bar{a}$ be the series $\langle a_1, a_2, \dots, a_n \rangle$. Let $\bar{s}$ be the series $\langle s_1, s_2, \dots, s_n \rangle$ obtained by sorting $\bar{a}$. The **sorting distance** of an element a_i is defined to be the distance between the position of a_i in $\bar{a}$ (which is i) and the position of a_i in $\bar{s}$, and is denoted as $d(a_i)$. For example if a_i is the smallest element in $\bar{a}$, then its position in $\bar{s}$ is 1 and $d(a_i) = i - 1$. The **sorting distance** of a series $\bar{a}$ is $d(\bar{a}) = \sum_{i=1}^{n} d(a_i)$. Prove or disprove the following statements.

1. Let $\bar{a}$ be the input of a comparison network and $\bar{b}$ be its output. For each i, it is the case that $d(a_i) \geq d(b_j)$ where j is the position of a_i in $\bar{b}$.

2. Let $\bar{a}$ be the input of a comparison network and $\bar{b}$ be its output. It is the case that $d(\bar{a}) \geq d(\bar{b})$.

Exercise 7 Let N be a sorting network, prove that for each i the top i wires and their comparators are a sorting network.

Exercise 8 Binary gcd:

1. Let a and b be even. What can be said regarding the connection between $gcd(a, b)$ and $gcd(a/2, b/2)$?

2. Let a be even and b odd. What can be said regarding the connection between $gcd(a, b)$ and $gcd(a/2, b)$?

3. Let a and b be odd. What can be said regarding the connection between $gcd(a, b)$ and $gcd(a - b, b)$?

Use the above to define a new *gcd* algorithm that only uses multiplication and division by 2. Analyze the complexity of the algorithm (assume multiplication and division by 2 can be done in constant time).

Exercise 9 The public key of Alice in the RSA encryption scheme is (n, e). Carl (which is no friend of Alice) has full access to the (encrypted) messages that are being sent once in a while to Alice. Carl has also designed a random algorithm A, that is able to decrypt 1% of the messages sent to Alice. That is, on 1% of the messages $M \in Z_n$ it is the case that $A(E(M)) = M$.

Alice is aware of Carl's algorithm, but is not worried as only 1% of the messages sent to her are known to Carl. Show that Alice should indeed worry, as Carl can design an efficient algorithm B (using his original algorithm A) that works on 99% of the messages M.

Hint: First prove that the RSA scheme is multiplicative. That is, $E(M_1)E(M_2) =_n E(M_1M_2)$.

Exercise 10 Let $n = pq$ where p and q are prime numbers. Prove:

1. For every $x \in Z_n$, x has a square root $(mod\ n)$ iff it has a square root $(mod\ p)$ and $(mod\ q)$.
2. If x has a square root $(mod\ n)$, then it has 4 different roots.
3. Finding the square root of x in Z_n is (essentially) as difficult as factoring n.

Exercise 11 Alice and Bob both have an n bit string: m_A and m_B respectively. Carl would like to know if they both have the same string. In the suggested setting both Alice and Bob can send a single message to Carl. Define the **communication complexity** of a protocol as the total number of bits sent during the execution of the protocol. Consider the following protocol:

1. Alice and Bob both treat their string as a $\sqrt{n} \times \sqrt{n}$ matrix. Denote Alice's string by $A[i][j]$ and Bob's string by $B[i][j]$.
2. Alice chooses a random row r_A between 1 and $\sqrt{n}$ and sends the entire row and the index r_A to Carl (*i.e.* Alice sends $A[r_A][*]$ and r_A to Carl).
3. Bob chooses a random column r_B between 1 and $\sqrt{n}$ and sends the entire column and the index r_B to Carl (*i.e.* Bob sends $B[*][r_B]$ and the index r_B to Carl).
4. Carl checks if $A[r_A][r_B] = B[r_A][r_B]$. If so then answers that the strings of Alice and Bob are identical, otherwise answers that they differ.

Answer the following questions:

1. If $m_A = m_B$, what will Carl answer after running the protocol.
2. If the strings differ on at least $n/2$ bits, with what probability will Carl answer correctly after running the protocol.
3. Let P_k be the k'th prime number (*e.g.* $P_1 = 2$, $P_5 = 11$). For $k = 1$ to $2n$ let $m_A[k] = m_A\ mod\ P_k$, and $m_B[k] = m_B\ mod\ P_k$. Show that:
   (a) If $m_A = m_B$ then for each k above $m_A[k] = m_B[k]$.
   (b) If $m_A \neq m_B$ then for at least n of the k's above $m_A[k] \neq m_B[k]$.
4. Suggest a protocol (with low communication complexity) in which the probability for Carl to be mistaken is bounded by 1/2 for **every** two strings m_A and m_B that differ.

Exercise 12 Let T be a given text and P be a given pattern. Show how to build an automata A that finds all occurrences of P in T that appear in odd shifts of T only.

Exercise 13 Let T be a given text and P be a given pattern. Two occurrences of P in T overlap if the first appears in an s_1 shift of T, and the second in an s_2 shift of T where $|s_1 - s_2| < |P|$. Suggest an algorithm which runs in linear time that, given T and P, finds the number of occurrences of P in T that overlap with a previous occurrences of P.

Exercise 14 Let T be a given text. Suggest an algorithm that finds the largest string X such that:

1. $T = XYX^R$, where Y is an arbitrary string, and X^R is the string X in reverse order (*e.g.* if $X = abc$ then $X^R = cba$).
2. $T = XYX$, where Y is an arbitrary string.

Exercise 15 Let T,T' be two given texts of identical length. Suggest an algorithm that decides if T is a cyclic permutation of T'.

Exercise 16 Let P and Q be two sets of n points each in the plane. It is known that there exists a line l which separates P and Q (that is all the points of P lie on one side of l and all the points of Q lie on the opposite side). Suggest an algorithm that find such a line l.

Exercise 17 Let P be a set of n points in the plane. Suggest an algorithm that constructs a simple polygon (one in which edges do not cut) that passes through all the points in P. Give a lower bound for the running time of such an algorithm (assuming a computational model in which sorting takes time $\Omega(n \log n)$).

Exercise 18 Let G be a connected planar graph on n vertices of maximal degree 3, and $\pi(G)$ be its embedding in the plane. It is well known that the number of edges and faces in $\pi(G)$ is linear. That is $\pi(G)$ is a collection of n points and $O(n)$ line segments (edges between these points) that form a partition of $R2$ into $O(n)$ faces. Assuming one can copy an array of size n in constant time, suggest an algorithm that: (a) Runs in time $O(n \log n)$. (b) Builds a data structure D of size $O(n2)$ which given any point p in $R2$ returns in time $O(\log n)$ the face f of $\pi(G)$ containing p.

Exercise 19 Let $P = \{p_1, \ldots, p_n\}$ be a set of n points that are sorted by their x-coordinate (*i.e.* for $i > j$ if $p_i = (p_i^x, p_i^y)$ and $p_j = (p_j^x, p_j^y)$ then $p_i^x \geq p_j^x$). Suggest an efficient algorithm for finding the convex hull of P.

Exercise 20 Let P be a polygon. Suggest an efficient algorithm for finding the convex hull of P.