DAA 2007: Mid-Term Exam II

[Answer **any 4** among the following 5 questions. **Total marks: 20**]

Q1.[5] Suppose 1 U.S. dollar buys 45 rupees, 1 rupee buys 2.5 Japanese yen, and 1 Japanese yen buys 0.01 U.S. dollars. Then, by converting currencies, a trader can start with 1 U.S. dollar and buy $45 \times 2.5 \times 0.01 = 1.125$ U.S. dollars, thus making a profit of 0.125 U.S. dollars.

Suppose we are given n currencies $c_1, c_1, \ldots, c_n$ and an $n \times n$ table R of exchange rates, such that one unit of currency c_i buys $R[i, j]$ units of currency c_j. Give an efficient algorithm to determine whether or not there exists a sequence of currencies $\langle c_{i_1}, c_{i_2}, \ldots, c_{i_k} \rangle$ such that

$$R[i_1, i_2] \cdot R[i_2, i_3] \cdots R[i_{k-1}, i_k] \cdot R[i_k, i_1] > 1.$$

Analyze the running time of your algorithm.

Q2.[5] On most computers, the operations of subtraction, testing if a number is even or odd, and halving can be performed more quickly than computing remainders. Let us design a gcd algorithm for two integers a and b keeping these points in mind.

(i) Prove that if a and b are both even, then $\gcd(a, b) = 2\gcd(a/2, b/2)$.

(ii) Prove that if a is odd and b is even, then $\gcd(a, b) = \gcd(a, b/2)$.

(iii) Prove that if a and b are both odd, then $\gcd(a, b) = \gcd((a-b)/2, b)$.

Using the above properties, design an efficient gcd algorithm for input integers a and b, where $a \geq b$, that runs in $O(\log a)$ time. Assume that each subtraction, odd/even test, and halving can be performed in unit time.

Q3.[5] Alice has a copy of a long n-bit file $A = \langle a_{n-1}, a_{n-2}, \ldots, a_0 \rangle$, and Bob similarly has an n-bit file $B = \langle b_{n-1}, b_{n-2}, \ldots, b_0 \rangle$. Alice and Bob wish to know if their files are identical. Alice can send her file A to Bob and Bob can compare A and B and check if their files are the same or not, and convey the yes/no answer to Alice. However this involves $n+1$ bits of communication, which is expensive. Alice and Bob want to send just $\Theta(\log n)$ bits and know the correct answer with a high probability.

So they use the following probabilistic check. Together, they select a prime $q > 1000n$ and randomly select an integer $x \in \{0, 1, \ldots, q-1\}$. Then Alice evaluates

$$A(x) = (\sum_{i=0}^{n-1} a_i x^i) \bmod q$$

and sends $A(x)$ to Bob. Bob similarly evaluates $B(x)$ and checks if $A(x) = B(x)$ or not. Prove that if $A \neq B$, then with probability at least 0.999 we have $A(x) \neq B(x)$ and if $A = B$, then $A(x)$ is necessarily the same as $B(x)$.

Hint. Use the following fact: We say that $\alpha \in Z_q$ is a *zero* of a polynomial $P(x)$ if $P(\alpha) = 0 \pmod{q}$. *A polynomial $P(x)$ of degree n has at most n distinct zeroes.*

Q4.[5] Consider two sets A and B each having n integers in the range 0 to $10n$. We wish to compute the *Cartesian sum* of A and B defined by

$$C = \{x + y : x \in A \text{ and } y \in B\}.$$

Note that integers in C are in the range from 0 to $20n$. We want to find the elements in C and the number of times each element in C is realized as a sum of elements of A and B. Show that the problem can be solved in $O(n \log n)$ time.

Q5.[5] Is the height of an n-node Fibonacci heap always $O(\log n)$? Why/why not?