CS 170 HW 11

Due on 2018-04-16, at 11:59 pm

1 (★) Study Group

List the names and SIDs of the members in your study group.

2 (★★★) Independent Set Approximation

In the Max Independent Set problem, we are given a graph $G = (V, E)$ and asked to find the largest set $V' \subseteq V$ such that no two vertices in V' share an edge in E.

Given an undirected graph $G = (V, E)$ in which each node has degree $\leq d$, show how to efficiently find an independent set whose size is at least $1/(d+1)$ times that of the largest independent set.

3 (★★★) Modular Arithmetic

(a) What is the last digit (i.e., the least significant digit) of 3^{4001}?

(b) Prove that for integers a_1, b_1, a_2, b_2, and n, if $a_1 \equiv b_1 \mod n$ and $a_2 \equiv b_2 \mod n$, then $a_1 \cdot a_2 \equiv b_1 \cdot b_2 \mod n$

(c) As in the last problem, show that if $a_1 \equiv b_1 \mod n$ and $a_2 \equiv b_2 \mod n$, then $a_1 + a_2 \equiv b_1 + b_2 \mod n$

(d) Give a polynomial time algorithm for computing $a^{b^c} \mod p$ for prime p and integers a, b, and c.

4 (★★★★) Wilson's Theorem

Wilson's theorem says that a number N is prime if and only if

$$(N-1)! \equiv -1 \pmod{N}.$$

(a) If p is prime, then we know every number $1 \leq x < p$ is invertible modulo p. Which of these numbers are their own inverse?

(b) By pairing up multiplicative inverses, show that $(p-1)! \equiv -1 \pmod{p}$ for prime p.

(c) Show that if N is *not* prime, then $(N-1)! \not\equiv -1 \pmod{N}$. [*Hint:* Consider $d = gcd(N, (N-1)!)$]

(d) Unlike Fermat's Little theorem, Wilson's theorem is an if-and-only-if condition for primality. Why can't we immediately base a primality test on this rule?

5 (★★) Random Prime Generation

Lagrange's prime number theorem states that as x increases, the number of primes less than x is approximated by $x/(log(x))$. Such abundance makes it simple to generate a random n-bit prime:

- Pick a random n-bit number N.
- Run a primality test on N.
- If it passes the test, output N; else repeat the process.

Show that this algorithm will sample on average $O(n)$ random numbers before hitting a prime. (Hint: If p is the chance of randomly choosing a prime and E is the average number of coin tosses, show that $E = 1 + (1 - p)E$)

Notice that this algorithm is different from other random algorithms we've seen, in that the randomness is in the runtime and not the correctness; It always returns a correct answer, but might take a long time to do so. Algorithms of this form are called *Las Vegas Algorithms*.

6 (★★★★) Quantum Gates

(a) The Hadamard Gate acts on a single qubit and is represented by the following matrix:

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Verify that this gate maps the basis states $|0\rangle$ and $|1\rangle$ to a superposition state that will yield 0 and 1 with equal probability, when measured. In other words, explicitly represent the bases as vectors, apply the gate as a matrix multiplication, and explain why the resulting vector will yield 0 and 1 with probabilities 1/2 each, when measured.

(b) Give a matrix representing a *NOT* gate. As in the previous part, explicitly show that applying your gate to the basis state $|0\rangle$ will yield the state $|1\rangle$ (and vice-versa).

(c) Give a matrix representing a gate that swaps two qubits. Explicitly show that applying this matrix to the basis state $|01\rangle$ will yield the state $|10\rangle$. Verify that this matrix is its own inverse.