CS 170 HW 12

Due on 2018-11-18, at 9:59 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 $\leq d$, give an efficient algorithm that finds an independent set whose size is at least $1/(d + 1)$ times that of the largest independent set. Only the main idea and the proof that the size is at least $1/(d + 1)$ times the largest solution's size are needed.

3 (★★) Modular Arithmetic

(a) Prove that for integers a_1, b_1, a_2, b_2, and n, if $a_1 \equiv b_1 \pmod{n}$ and $a_2 \equiv b_2 \pmod{n}$, then $a_1 \cdot a_2 \equiv b_1 \cdot b_2 \pmod{n}$.

(b) As in the last problem, show that if $a_1 \equiv b_1 \pmod{n}$ and $a_2 \equiv b_2 \pmod{n}$, then $a_1+a_2 \equiv b_1 + b_2 \pmod{n}$.

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

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 Wilson's theorem?

5 (★★) Random Prime Generation

Lagrange's prime number theorem states that as N increases, the number of primes less than N is $\Theta(N/\log(N))$. Consider the following algorithm for choosing a random n-bit prime.

- Pick a random n-bit number k.
- Run a primality test on k.
- If it passes the test, output k; 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 expected number of random samples, show that $E = 1 + (1 - p)E$.)

6 (★★★) Streaming Algorithms

In this problem, we assume we are given an infinite stream of integers x_1,x_2, . . . , and have to perform some computation after each new integer is given. Since we may see many integers, we want to limit the amount of memory we have to use in total. For all of the parts below, give a brief description of your algorithm and a brief justification of its correctness.

(a) Show that using only a single bit of memory, we can compute whether the sum of all integers seen so far is even or odd.

(b) Show that we can compute whether the sum of all integers seen so far is divisible by some fixed integer N using $O(\log N)$ bits of memory.

(c) Assume N is prime. Give an algorithm to check if N divides the product of all integers seen so far, using as few bits of memory as possible.

(d) Now let N be an arbitrary integer, and suppose we are given its prime factorization: $N = p_1^{k_1} p_2^{k_2} \dots p_r^{k_r}$. Give an algorithm to check whether N divides the product of all integers seen so far, using as few bits of memory as possible. Write down the number of bits your algorithm uses in terms of $k_1, \dots, k_r$.