Theory of Computation

Final Exam, 2016 Fall Semester,
1/10/2017

Note: Unless stated otherwise, you may use any results proved in class

Problem 1 (25 points) For the Diffie-Hellman Secret-Key Agreement Protocol, Alice and Bob agree on a large prime p and a primivite root g of p (where p and g are public). Alice chooses a random a and Bob also chooses a random b. For $p = 23$, $g = 5$, $a = 6$ and $b = 15$, what are the values of α, β and the common key?

Ans:
For $p = 23$, $g = 5$, $a = 6$ and $b = 15$, the values of α and β are

$$\alpha \equiv 5^6 \equiv 8 \pmod{23},$$
$$\beta \equiv 5^{15} \equiv 19 \pmod{23},$$

and the common key is

$$\alpha^b \equiv 8^{15} \equiv \beta^a \equiv 2 \pmod{23}.$$

■

Problem 2 (25 points) Prove that if every language in BPP only needs a pseudorandom generator which stretches a random seed of logarithmic length, then BPP = P.

Ans:
We only need to show BPP $\subseteq$ P. Run the BPP algorithm for each of the seeds. There are only $2^{O(\log n)} = O(n^c)$ seeds, a polynomial Accept if and only if at least 3/4 of the outcomes is a "yes." The running time is deterministically polynomial. ■

Problem 3 (25 points) Let $n \in \mathbb{Z}^+$ with $n \geq 2$. Let $\phi(n)$ stand for Euler's totient function, which counts the number of positive integers smaller than n and are relatively prime to n.

1. (5 points) Determine $\phi(2^n)$.
2. (10 points) Determine $\phi(\phi(2^n))$.
3. (10 points) Determine $\phi\left((2p)^n\right)$ where p is an odd prime.

Ans:

1. $\phi\left(2^n\right) = 2^n - 2^{n-1} = 2^{n-1}(2-1) = 2^{n-1}$.

2. $\phi\left(\phi(2^n)\right) = \phi\left(2^{n-1}\right) = 2^{n-1} - 2^{n-2} = 2^{n-2}(2-1) = 2^{n-2}$.

3. $\phi\left((2p)^n\right) = \phi\left(2^n p^n\right) = \phi\left(2^n\right)\phi\left(p^n\right) = 2^{n-1}(p^n - p^{n-1}) = 2^{n-1}p^{n-1}(p-1)$.

$\blacksquare$

Problem 4 (25 points) Prove that there is no ϵ-approximation algorithm for the NP-complete 6-COLORING if $\epsilon < 1/7$ and assuming P $\neq$ NP. Recall that an ϵ-approximation algorithm F guarantees that

$$OPT \leq c(F(G)) \leq \frac{OPT}{1-\epsilon}$$

where $c(F(G))$ is the number of colors the polynomial-time algorithm F uses to color G.

Ans:
We prove the problem by contradiction. We assume that there exists an ε-approximation algorithm F that colors the graph G in polynomial time. Given $\epsilon < 1/7$, F will color G with at most

$$x = \frac{OPT}{1-\epsilon} = 6$$

in polynomial time if G is 6-colorable. That is, F can answer *YES* or *NO* to the NP-complete problem 6-COLORING in polynomial time. $\blacksquare$