CS670 Final

Rules

1. Due time: 8 am, Tuesday, May 9, 2023.
2. You must turn in your work on or before the due time.
3. If you are asked to construct an algorithm. You should justify its correctness and analyze its running time.
4. The final set you turn in should be typed-up, using Latex or Word point size 11 or 12.
5. You may consult the textbook, lecture slides and notes, homework solutions, and quote as facts from these sources when solving a problem. You may not consult any other source.
6. You must work independently without collaborating with others on the exam set. You must write up the solution on your own and by yourself.
7. Three questions, 10 points each. Be concise and accurate.

Academic Integrity Honor Code Pledge

I pledge to uphold the highest academic standards and integrity. In accordance with USC Viterbi's Honor Code (https://viterbischool.usc.edu/academic-integrity/), I affirm that I have not used any unauthorized materials in completing this exam, and have neither given assistance to others nor received assistance from others. Further, I affirm that I have not observed any other students in this class acting to gain an unfair advantage, or I have reported to my instructor any activity I have observed that is not in accordance with USC Viterbis Honor Code. I do so to sustain a Viterbi culture of integrity, responsibility, community and excellence in all our endeavors. I understand that there are significant consequences for violating academic integrity (https://policy.usc.edu/scampus-part-b/) and that suspected violations will be reported to the School and the University.

Problems

1. Consider the following problem: Given A, t, where A is a finite set of positive integers, and t is a positive integer, we would like to find a subset $S \subseteq A$ such that the subset sum is as large as possible but bounded by t, that is $\sum_{x \in S} x$ is maximum possible subject to the condition that $\sum_{x \in S} x \leq t$.

   (a) Show that the problem is NP-hard by reducing from the Subset-Sum problem.

   (b) Show that the maximization problem has polynomial time $\frac{1}{2}$-approximation algorithm. In other word, there is a polynomial time algorithm which on input A, t as described above, outputs a subset S such that $z = \sum_{x \in S} x \leq t$ and $z \geq \frac{1}{2} t^*$ where $t^* = \max\{\sum_{x \in B} x : B \subseteq A, \sum_{x \in B} x \leq t\}$, the maximum subset sum that is upper bounded by t.

2. Suppose for every $L \in NP$ there is a polynomial-time transformation f with the following property:

   if $x \in L$ then $f(x)$ is a graph which contains a clique on k vertices where $k \geq \frac{2}{3} n$ and n is the number of vertices of G;

   if $x \notin L$ then $f(x)$ is a graph which contains no clique on k vertices where $k \geq \frac{1}{3} n$ and n is the number of vertices of G.

   Show that from the above assumption it would follow that if $P \neq NP$ then there is no polynomial-time $\frac{1}{2}$-approximation algorithm for the Max-Clique problem, which on input a graph G finds a maximum clique that G contains as a subgraph.

3. For all positive integers n, let $H_n := \{a \in \mathbb{Z}_n^* : a = \alpha^2 \mod n \text{ for some } \alpha \in \mathbb{Z}_n^*\}$. We call α a square root of a in $\mathbb{Z}_n^*$ if $a \equiv \alpha^2 \mod n$. We say that n is an *RSA number* if $n = pq$ where p and q are two distinct odd prime numbers.

   (a) Suppose n is an RSA number with $n = pq$, p and q being odd prime numbers. Show that under the projection map $\mathbb{Z}_n^* \to Z_p^* \times \mathbb{Z}_q^* : x \to (x \mod p, x \mod q)$, H_n maps bijectively to $H_p \times H_q$. Moreover suppose $a \in H_n$ and $W \subseteq H_n$. Then for uniform random $r \in Z_n^*$, the probability that $r^2 a \in W$ is $\frac{|W|}{|H_n|}$.

   (b) Suppose there is a polynomial-time algorithm A which on input an RSA number n and $a \in H_n$, successfully finds a square root of a in $\mathbb{Z}_n^*$ on one percent of $a \in H_n$. Show that using A as a subroutine we can construct a probabilistic polynomial-time algorithm that on input an RSA number n and $a \in H_n$ always finds a square root of a in $\mathbb{Z}_n^*$.