(CS1.406) Advanced Algorithms (Spring 2022) Due: 07.03.2022

Problem Set 3

Instructions:

- Discussions amongst the students are not discouraged, but all writeups must be done individually and must include names of all collaborators.
- Referring sources other than the lecture notes is discouraged as solutions to some of the problems can be found easily via a web search. But if you do use an outside source (eg., text books, other lecture notes, any material available online), do mention the same in your writeup. This will not affect your grades. However dishonesty of any sort when caught shall be heavily penalized.
- Be clear in your arguments. Vague arguments shall not be given full credit.

1. Recall the bit-fixing scheme for *Oblivious Permutation Routing* from Lecture 8.

   Let G be a hypercube on n bits. For vertices $u, v \in \{0,1\}^n$, bit-fixing scheme decides a path from $u \rightsquigarrow v$ by scanning the bit strings corresponding to u and v from the most significant bit to the least significant bit and flips the individual bits in order. That is, correct the mismatched bits in order.

   Now consider the following variant of the bit-fixing scheme. Each packet p_v (for $v \in \{0,1\}^n$) randomly orders the position of bits in the string of its source v and corrects the mismatched bits in order (of this new random string). Show that there is a permutation for which with high probability this algorithm requires $2^{\Omega(n)}$ steps to complete the routing. [10 marks]

2. Recall the problem of MAXSAT that we discussed in Lecture 9.

   Given a set of m clauses over n variables, we ask what is the maximum number of satisfiable clauses, over all the assignments $\{T, F\}^n$ to the variables. In class we showed a random assginment that satisfies at least $\frac{m}{2}$ clauses in expectation. Give a deterministic algorithm to give an assignment that satisfies at least $\frac{m}{2}$ clauses. [6 marks]

3. Let us recall the problem of set-balancing from Lecture 5.

   Let A be a $n \times m$ matrix with entries that are either 0 or 1. Let $\mathbf{b} \in \{-1,1\}^m$, $\mathbf{c} \in \mathbb{Z}^n$ and

   $$A \cdot \mathbf{b} = \mathbf{c}.$$

   We are looking for a vector $\mathbf{b} \in \{-1,1\}^m$ that minimizes

   $$\|A \cdot \mathbf{b}\|_\infty = \max_{i \in [n]} \{c_i\}.$$

   In class, we saw that when each entry of $\mathbf{b}$ was chosen independently and uniformly at random from $\{-1,1\}$ it can be proved that with a probability of at least $1 - \frac{2}{n}$, each c_i has a value of at most $\sqrt{4m \ln n}$. Using the method of conditional expectations, can we find an assignment such that each c_i has a value of at most $\sqrt{4m \ln n}$. [10 marks]

4. Let m, n be large integers such that $m > n$. For an integer q, $\mathbb{Z}_q$ is the set of integers modulo q with the operations of addition and multiplication.

   Let the problem $\mathrm{Max} - 2\mathrm{Lin}(\mathbb{Z}_q)$ be defined as follows – Given a system S of m linear equations over n variables, each of which depends on at most two variables, what is the maximum number of linear equations that can be satisfied over all the assignments to the variables from $\mathbb{Z}_q$?

   Formally, let $X = \{x_1, \ldots, x_n\}$ and $S = \{L_1(X) = 0, \ldots, L_m(X) = 0\}$. For all $i \in [m]$, each $L_i(X) = 0$ is of the form

$$c_{i_1}x_{i_1} + c_{i_2}x_{i_2} + c_{i_3} \equiv 0 \mod q \quad \text{for some constants } c_{i_1}, c_{i_2} \text{ and } c_{i_3}\,.$$

   (a) Provide a randomized scheme for the assignment of the variables such that the expected number of linear equations that can be satisfied is $\frac{m}{q}$.

   (b) Show that we can deterministically obtain an assignment that satisfies at least $\frac{m}{q}$ linear equations in S.

[6+8 marks]

5. Let n be an asymptotically large integer. Using the probabilistic method[1] show that there is a bipartite graph $G = (L, R, E)$ with the following properties.

   - $|L| = |R| = n$.
   - Each vertex in L has a degree of $n^{3/4}$ and each vertex in R has a degree of at most $3n^{3/4}$.
   - Every subset of $n^{3/4}$ vertices in L has at least $n - n^{3/4}$ neighbours in R.

[15 marks]

6. Let there be n medical schools $\{M_1, \ldots, M_n\}$ each with exactly 1 vacancy for a medical residency program and let there be n medical students $\{S_1, \ldots, S_n\}$ applying for those positions. Each medical school has a prior knowledge of all the students who are applying and thus has a (strict) preference[2] amongst them. Similarly, each student has a strict preference for the medical schools. A matching in this case is the assignment of these n medical students to the n vacancies across all the medical schools. A matching is said to be unstable if there is a medical school M_i, and a student S_j such that

   - S_j is not matched with M_i,
   - S_j prefers M_i to their current match, and
   - M_i prefers S_j to their current match.

   A matching is said to be stable if it is not unstable. Gale and Shapley proposed Algorithm 1.

   (a) Prove that whenever there is an unmatched student, there is a medical school that they have not approached.

   (b) Algorithm 1 terminates in $O(n^2)$ steps.

   (c) The matching thus found by Algorithm 1 is stable.

[1] Refer to pages 1–3 in the notes of Lecture 9

[2] A preference in this case means that each medical school rates the students in a strict decreasing order of desirability for the position.

(d) Let us modify Line 3 of Algorithm 1 to the following – Student S_i picks a medical school uniformly at random among all n schools (including those they were rejected from), and Lines 9 and 12 are deleted. What is the expected running time to get a stable matching.

[5+5+5+15 marks]

Algorithm 1: GaleShapleyProposalAlgorithm

```
1  while there is an unmatched student do
2      Let S_i be an unmatched student;
3      S_i approaches a medical school M_j that they most prefer which they were not already
         rejected from;
4      if M_j's vacancy is open then
5          M_j selects student S_i;
6      else
7          if S_i is more prefered by M_j than their current selection then
8              M_j selects S_i and unmatches their current selection;
9              S_i' adds M_j into the list of schools they were rejected from;
10         else
11             S_i remains unmatched;
12             S_i adds M_j into the list of schools they were rejected from;
13         end
14     end
15 end
```