Algebra and Computation: Midsem

Instructions

1. The problem set has **1 question (two subdivisions)** with a total score of **40 points**.
2. You may request a hint for the two subquestions. The hint for (i) will cost **5 points** and hint for (ii) will cost **10 points**.
3. You are **not allowed** to discuss with other classmates.
4. You are allowed to consult your notes and/or books relevant, but **not allowed** to use your personal electronic devices. Should you want to consult the online lecture notes from last time, it would be available in the machine in the seminar room (please avoid printing the whole thing).

Question 1. Suppose we have some group $G \leq S_n$. We have been seeing the following very natural tower:

$$G = G^{(0)} \geq G^{(1)} \geq \cdots \geq G^{(n)} = \text{id},$$

where recall that $G^{(i)} = \{g \in G : j^g = j \text{ for all } j \leq i\}$ (the point-wise stabilizer of the first i elements). The following is a very useful definition when it comes to computational group theory.

Definition (Strong Generating Set)**.** *A generating set S of G is said to be a* strong generating set (SGS) *if $S \cap G^{(i)}$ is a generating set of $G^{(i)}$ for all i.* ◊

(i) **[10 points]** Given a group $G \leq S_n$ as $G = \langle T \rangle$, give a polynomial time algorithm to construct a *strong generating set* of size at most $n(n-1)/2$.

(ii) **[30 points]** Use this idea to give a formal proof of the following that was alluded to in class but not proved.

> Suppose we are provided an oracle GraphIso that, given two graphs $X = (V_1, E_1)$ and $Y = (V_2, E_2)$ as inputs tells you whether the graphs are isomorphic or not. That is, someone has provided a library that somehow magically computes this function GraphIso.
>
> Using this oracle (by making poly(n)-many adaptive calls to it if necessary), show how you get a poly(n)-time algorithm for the GraphAut problem where you are given a graph X and you want to compute a generating set of the automorphism group of it.