

## Problem Set 6

Out: February 21

Due: February 28

Reminder: you are encouraged to work in groups of two or three; however you must turn in your own write-up and note with whom you worked. You may consult the course notes and the text (Sipser). The full honor code guidelines and collaboration policy can be found in the course syllabus.

Please attempt all problems. Please select *one* the first two problems to be graded completely (and indicate clearly which one); the other one will receive 1 point for a credible attempt. **Please turn in your solutions via Gradescope, by 1pm on the due date.**

1. Recall that two graphs  $G_1 = (V_1, E_1)$  and  $G_2 = (V_2, E_2)$  are *isomorphic* if there is a bijection  $f : V_1 \rightarrow V_2$  such that  $(u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$ . On the Problem Set 4 you showed that  $\text{CONTAINS}_H$  is in P.

Prove that the following variation in which  $H$  is part of the input is NP-complete:

$$\text{SUBGRAPH ISOMORPHISM} = \{(G, H) : G \text{ contains a subgraph isomorphic to } H\}.$$

As in the previous problem set, by “subgraph” we mean a subset of  $G$ 's vertices together with all of  $G$ 's edges on that subset of vertices. A side note: the related problem

$$\text{GRAPH ISOMORPHISM} = \{(G, H) : G \text{ is isomorphic to } H\}$$

is not known to be NP-complete, nor is it known to be in P.

2. Let  $\mathcal{C} = \{S_1, S_2, S_3, \dots, S_n\}$  be a collection of sets and define  $U = \cup_i S_i$ . We say that  $T \subseteq \mathcal{C}$  is a *cover* if every  $u \in U$  occurs in at least one  $S_i \in T$ . Prove that the language:

$$\text{SET COVER} = \{(\mathcal{C}, k) : \text{there is a cover } T \subseteq \mathcal{C} \text{ with } |T| \leq k\}.$$

is NP-complete.

3. A *cut* of an undirected graph  $G = (V, E)$  is a subset of vertices  $S \subseteq V$ . We say that an edge *crosses the cut* if one endpoint is in  $S$  and the other is in  $V \setminus S$ . In class, we will show that the following language is NP-complete:

$$\text{MAXCUT} = \{(G = (V, E), k) : \text{there is a cut } S \subseteq V \text{ with at least } k \text{ edges crossing it}\}.$$

In this problem you will consider a related problem concerning special cuts called bisections. A *bisection* of a graph  $G = (V, E)$  is a cut  $S \subseteq V$  with  $|S| = |V|/2$ . A powerful and general algorithmic strategy is to represent the “components” of a problem by vertices of a graph  $G$  and dependencies between components by its edges; then a bisection splits the problem into two equal pieces, which can be solved recursively and merged to produce a global solution. In this general framework, the work in the merging step often depends on the number of

edges crossing the bisection. Thus, one might hope to be able to efficiently solve the following problem, MINIMUM BISECTION:

$\{(G, k) : G \text{ is a connected multigraph having a bisection with at most } k \text{ edges crossing it.}\}$

Unfortunately, an efficient solution is unlikely. Prove that MIN BISECTION is NP-complete.

Hint: what is the relationship between this problem and the maximization version, MAXIMUM BISECTION:

$\{(G, k) : G \text{ is a connected multigraph having a bisection with at least } k \text{ edges crossing it.}\}?$

4. In class we will show that the following problem concerning positive integers is NP-complete:

$$\text{SUBSET - SUM} = \left\{ (a_1, a_2, \dots, a_n, B) : \exists S \subseteq [n] \text{ for which } \sum_{i \in S} a_i = B. \right\}$$

This will be useful when giving reductions for the following problems:

- (a) A list of positive integers  $(a_1, a_2, \dots, a_n)$  is *partitionable* if there is a subset  $T \subseteq \{1, 2, \dots, n\}$  for which  $\sum_{i \in T} a_i = \sum_{i \notin T} a_i$ . Show that the following language is NP-complete:

$$\text{PARTITION} = \{S = (a_1, a_2, \dots, a_n) : S \text{ is partitionable}\}$$

- (b) In the “knapsack problem” we are given  $n$  items each of which has a positive integer *cost*  $c_i$  and a positive integer *value*  $v_i$ . We are asked to pack a knapsack with items whose total value is at least  $V$ , and without exceeding a budget  $C$ . Show that the following language is NP-complete:

$$\begin{aligned} \text{KNAPSACK} = & \{(c_1, c_2, \dots, c_n, v_1, v_2, \dots, v_n, V, C) : \exists T \subseteq \{1, 2, \dots, n\} \\ & \text{for which } \sum_{t \in T} v_t \geq V \text{ and } \sum_{t \in T} c_t \leq C\} \end{aligned}$$