# CSCI3160: Special Exercise Set 12

Prepared by Yufei Tao

**Problem 1 (Textbook Exercise 35.3-1).** Consider $\mathcal{S} = \{$`arid`, `dash`, `drain`, `heard`, `lost`, `nose`, `shun`, `slate`, `snare`, `thread`$\}$. Treat each word in $\mathcal{S}$ as a set of letters. Run the set-cover algorithm discussed in the lecture and describe its output.

**Problem 2.** Recall that our set-cover algorithm in each iteration picks a set with the largest *benefit*. Prove: if we lay out the sets in the order they are picked, their benefits are non-ascending.

**Problem 3*.** Give an example to show that the approximation ratio of our set-cover algorithm cannot be bounded by 2.

**Problem 4.** Consider the set $\mathcal{S}$ given in Problem 1. This time, we want to treat it as an input to the hitting set problem. Re-formulate the problem as a set-cover problem.

**Problem 5.** Let $\boldsymbol{M}$ be an $n \times m$ matrix where each cell is either 0 or 1. It is guaranteed that every row of $\boldsymbol{M}$ has at least one 1. A set $S$ of columns is a *column cover* if every row of $\boldsymbol{M}$ has a 1 in at least one column of $S$. If OPT is the minimize size of all column covers, describe a poly$(n, m)$-time algorithm (i.e., polynomial in $n$ and $m$) that finds a column cover of size $O(\text{OPT} \cdot \log n)$.