### Challenge Problems 6

due 12/7/2022 at 11:59pm in Gradescope

**Instructions.** Limited collaboration is allowed while solving problems, but you must write solutions yourself. List collaborators on your submission.

You can choose which problems to complete, but must submit at least one problem per assignment. See the course page for information about how challenge problems are graded and contribute to your homework grade. Since you don’t need to complete every problem, you are encouraged to focus your efforts on producing high-quality solutions to the problems you feel confident about. There is no benefit to guessing or writing vague answers.

If you are asked to design an algorithm, please (a) give a precise description of your algorithm using either pseudocode or language, (b) explain the intuition of the algorithm, (c) justify the correctness of the algorithm; give a proof if needed, (d) state the running time of your algorithm, (e) justify the running-time analysis.

**Submissions.** Please submit a PDF file. You may submit a scanned handwritten document, but a typed submission is preferred. Please assign pages to questions in Gradescope.

This challenge set has extra problems from previous chapters that you may attempt to try to meet your goals for challenge problems. Course staff won’t provide much support on extra problems. TAs will prioritize questions about intractability problems, and may not be familiar with extra ones.

## 1 Challenge Problems about Intractability

### Problem 1. $k$-coloring.

We say that a graph $G = (V, E)$ is $k$-colorable, if there exists a coloring $\phi : V \rightarrow \{1, \ldots, k\}$ such that for every $(u, v) \in E$, $\phi(u) \neq \phi(v)$. In words, we can color all of the vertices in the graph, using at most $k$ colors, so that no neighboring vertices have the same color. For example, the graph below is 3-colorable but it is not 2-colorable.

![Graph](image)

In the $k$-COLORING problem we are given a graph $G = (V, E)$ and our goal is decide whether the graph is $k$-colorable.

Prove that $3$-COLORING $\leq_P$ INDEPENDENTSET.

### Problem 2. Summer Camp Recruiting.

Suppose you’re helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who’s skilled at each of the $n$ sports covered by the camp (baseball, volleyball, and so on). They have received job applications from $m$ potential counselors. For each of the $n$ sports, there is some subset of the $m$ applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most $k$ of the counselors and have at least one counselor qualified in each of the $n$ sports? We’ll call this the Efficient Recruiting Problem.

Show that Efficient Recruiting is NP-complete.

### Problem 3. Hitting set.

Consider a set $A = \{a_1, \ldots, a_n\}$ and a collection $B_1, \ldots, B_m$ of subsets of $A$ (i.e., $B_i \subseteq A$ for all $i$). We say that $H \subset A$ is a hitting set for the collection if $H$ contains at least one element from each $B_i$, that is $H \cap B_i$ is non-empty for all $i$ (so $H$ “hits” all the sets $B_i$).
The Hitting-Set problem is the following: Given a set $A = \{a_1, \ldots, a_n\}$, subsets $B_1, \ldots, B_m \subset A$, and a number $k$, is there a hitting set $H \subset A$ of size at most $k$?

Prove that Hitting-Set is NP-Complete.

**Problem 4. Zero Weight Cycle.** You are given a directed graph $G = (V, E)$ with weights $w_e$ on its edges $e \in E$. The weights can be negative or positive. The Zero-Weight-Cycle Problem is to decide if there is a simple cycle in $G$ so that the sum of the edge weights on this cycle is exactly 0.

Prove that Zero-Weight-Cycle is NP-complete. (Hint: try a reduction from Subset-Sum.)

## 2 Challenge Problems from Previous Chapters

**Problem 5. Connectivity in Weighted Graphs (Chapter 4).** Let $G = (V, E, W)$ be a connected weighted graph where each edge $e$ has an associated non-negative weight $w(e)$. We call a subset of edges $F \subset E$ unseparating if the graph $G' = (V, E \setminus F)$ is connected. This means that if you remove all of the edges $F$ from the original edge set, this new graph is still connected. For a set of edges $E' \subset E$ the weight of the set is just the sum of the weights of the individual edges that is $w(E') = \sum_{e \in E'} w(e)$. Give a polynomial time algorithm to find the unseparating set with maximum weight.

**Problem 6. Split a string into words (Chapter 6).** You are given a string of $n$ characters $s[1\ldots n]$, which you believe to be a corrupted text document in which all punctuation has vanished (so that it looks something like “itwasthebestoftimes…”). You wish to reconstruct the document using a dictionary, which is available in the form of a Boolean function $dict$: for any string $w$,

\[
dict(w) = \begin{cases} 
  \text{true} & \text{if } w \text{ is a valid word}, \\
  \text{false} & \text{otherwise}.
\end{cases}
\]

1. Give a dynamic programming algorithm that determines whether the string $s$ can be reconstituted as a sequence of valid words. The running time should be at most $O(n^2)$, assuming calls to dict take unit time.

2. In the event that the string is valid, make your algorithm output the corresponding sequence of words.

**Problem 7. Holidays and employees (Chapter 7).** There is a set $H$ of holidays and a set $W$ of employees. On each holiday, exactly one employee must be at work. Each employee $w$ has a set $H_w \subset H$ of holidays when they can work and a maximum number $m_w$ of holidays that they can work. Give an efficient algorithm that assigns employees to holidays when they should work, or reports this is not possible.