Instructions: Please note that handwritten assignments **will not be graded**. Use the provided LaTeX template to complete your homework. Please do not alter the order or spacing of questions (keep each question on its own page). When you submit to Gradescope, you must mark which page(s) correspond to each question. **You may not receive credit for unmarked questions**.

When including graphical figures, we encourage the use of tools such as graphviz or packages like tikz for simple and complex figures. However, these may be handwritten only if they are neat and legible (as defined by the grader).

Important Homework Information

- Each homework assignment is worth 100 points.
- All homeworks are weighted evenly.
- Each homework will have some easier questions from the current week's content, and some harder questions from the previous week's content.
- It is recommended that you work on the questions from the previous week's content throughout the week. The questions related to the current week's content should be completed throughout the week.
- The point value of each question is listed next to the question.
- In brackets after the point amount are two values: the week number the question is related to, and a difficulty score in 1-5 ⋆s. These are meant to help you determine which problems will likely take more time. The scores are relative to the questions in that particular homework.
- Include one question per page. (It's ok to keep parts on the same page.)

1. For the following questions, select whether the statement is true or false, and write a *brief* explanation of your reasoning.

   (a) (10 points) [W0, ⋆] There will be a homework assignment most weeks.

   ■ True □ False

   (b) (10 points) [W0, ⋆] All homework assignments are due on Saturday night.

   ■ True □ False

   (c) (10 points) [W3, ⋆⋆⋆⋆⋆] This problem is marked as a problem about content from Week 3 with minimal difficulty.

   □ True ■ False

2. (70 points) [W0, ⋆ ⋆ ⋆] Design a dictionary data structure in which search, insertion, and deletion can all be processed in $\mathcal{O}(1)$ time in the worst case. You may assume the set elements are integers drawn from a finite set $1, 2, \ldots, n$, and initialization can take $\mathcal{O}(n)$ time.

    1. Allocate an array of length n.
    2. Search for key i can be performed by looking if an element exist at index i.
    3. Insert and remopve at key i is place/remove element at index i.
    4. Insert, remove, and index are all $\mathcal{O}(1)$. so all operations are constant time.