CSC165H1: Problem Set 4

Due December 4, 2019 before 4pm

General instructions

Please read the following instructions carefully before starting the problem set. They contain important information about general problem set expectations, problem set submission instructions, and reminders of course policies.

- Your problem sets are graded on both correctness and clarity of communication. Solutions which are technically correct but poorly written will not receive full marks. Please read over your solutions carefully before submitting them. Proofs should have headers and bodies in the form described in the course note.
- Each problem set may be completed in groups of up to three. If you are working in a group for this problem set, please consult `https://github.com/MarkUsProject/Markus/wiki/Student_Groups` for a brief explanation of how to create a group on MarkUs.

  Exception: Problem Sets 0 and 1 must be completed individually.
- Solutions must be typeset electronically, and submitted as a PDF with the correct filename. **Hand-written submissions will receive a grade of ZERO.**

  The required filename for this problem set is **problem_set4.pdf**.
- Problem sets must be submitted online through MarkUs. If you haven't used MarkUs before, give yourself plenty of time to figure it out, and ask for help if you need it! If you are working with a partner, you must form a group on MarkUs, and make one submission per group. "I didn't know how to use MarkUs" is not a valid excuse for submitting late work.
- Your submitted file should not be larger than 9MB. This may happen if you are using a word processing software like Microsoft Word; if it does, you should look into PDF compression tools to make your PDF smaller, although please make sure that your PDF is still legible before submitting!
- The work you submit for credit must be your own; you may not refer to or copy from the work of other groups, or external sources like websites or textbooks. You may, however, refer to any text from the Course Notes (or posted lecture notes), except when explicitly asked not to.

Additional instructions

- For questions in which you're analysing algorithms, you may use (without proving them) all of the theorems given in the **Properties of Big-Oh, Omega, and Theta** handout, as long as you clearly state which theorems you are using.

1. **[12 marks] algorithm analysis**

   Read over the code for **has_odd**. Assume that **number_list** contains entries from $\{0, 1, 2, 3, 4\}$, with duplicates allowed. Answer the questions below, assuming that $n = \textbf{len}(\text{number\_list})$

```python
def has_odd(number_list) -> bool:
  for i in range(len(number_list)):
    if number_list[i] % 2 == 1:
      return True
  return False
```

(a) **[3 marks]** Find a good upper bound, $U(n)$, for $WC_{has\_odd}(n)$. Prove that your upper bound is correct.

(b) **[3 marks]** Find a lower bound, $L(n)$, for $WC_{has\_odd}(n)$ that is in the same asymptotic complexity class as $U(n)$ (that's what I mean by "good" in the previous part). Prove that your lower bound is correct, then state and justify a simple big-Theta complexity class for $WC_{has\_odd}(n)$

(c) **[3 marks]** If **has_odd** returns **True** after examining k entries in **number_list**, count this as k steps. If an **has_odd** examines all n entries in **number_list** and proceeds to return **False** count this as $n + 1$ steps. Using these assumptions, show how to calculate the average number of steps for all inputs to **has_odd** of length 2.

(d) **[3 marks]** Use the step-counting assumptions in the previous part to devise a formula for the average number of steps for all inputs to **has_odd** of length n. You may find it useful to recall (where r is some positive real number)

$$\sum_{i=0}^{i=n-1} ir^i = \frac{nr^n}{r-1} + \frac{r - r^{n+1}}{(r-1)^2}$$

Show your work.

2. **[18 marks] graph connectivity**

Answer the questions below. Assume $|V|$ is finite and positive.

(a) **[3 marks]** Prove that for all undirected graphs $G = (V, E)$, if C is a cycle in G and e is an edge in C, then removing e leaves C connected. Notice that this is used in Example 6.8, so you cannot use Example 6.8 as proof, nor can you use the fact that this is asserted without proof in the paragraph just before Example 6.8.

(b) **[3 marks]** Prove or disprove: In every undirected graph $G = (V, E)$ with all vertices having degree at least $\lfloor |V|/3 \rfloor$, for every 3 distinct vertices $u, v, w \in V$ there is a path of length no more than 2 from u to v, or from v to w, or from w to u.

(c) **[3 marks]** Prove or disprove: If graph $G = (V, E)$ has an odd number of vertices with even degree, then $|V|$ is odd.

(d) **[3 marks]** Prove or disprove: Every undirected graph $G = (V, E)$ with at least 13 vertices, all vertices having degree at least $|V| - 7$, is connected.

(e) **[3 marks]** Prove that every undirected graph $G = (V, E)$ with every vertex having degree at least 5, has a cycle.

(f) **[3 marks]** Prove or disprove: If $G = (V, E)$ is an undirected graph where every vertex has degree at least 4 and $u \in V$, then there are at least 64 distinct paths in G that start at u.