

## CSC165H1: Problem Set 4

Due December 6, 2017 before 10pm

### 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](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

- All final Big-Oh, Omega, and Theta expressions should be fully simplified according to three rules: don't include constant factors (so  $\mathcal{O}(n)$ , not  $\mathcal{O}(3n)$ ), don't include slower-growing terms (so  $\mathcal{O}(n^2)$ , not  $\mathcal{O}(n^2 + n)$ ), and don't include floor or ceiling functions.
- You may use common algebraic properties of logarithms (change of base, log of a product, etc.), and the fact that for all  $x, y \in \mathbb{R}^+$ ,  $x > y \Leftrightarrow \log x > \log y$ .
- 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. [18 marks] **vertex degree** Recall the definition of the degree of a vertex:

**Definition 1** (degree of  $v$ ,  $d(v)$ ).  $d(v)$ :  $|\{(v, u) \mid (v, u) \in E\}|$ , for  $v \in V, G = (V, E)$

- (a) [3 marks] Prove that for any graph  $G = (V, E)$  the number of vertices with odd degree is even.  
 (b) [3 marks] Let  $G = (V, E)$  be a graph with 4 vertices and the following (incomplete) list of degrees:

- $d(v_1) = 1$
- $d(v_2) = 2$
- $d(v_3) = 3$

Provide a complete list of edges that could satisfy the above list of degrees as well as including vertex  $v_4$ . Prove that your solution is the only one possible.

- (c) [3 marks] Prove that every graph  $G = (V, E)$  that has at least one vertex  $v$  with degree  $d(v) = n$  must also have  $|V| \geq n + 1$ .  
 (d) [3 marks] Prove that every graph  $G = (V, E)$  where  $\forall v \in V, d(v) = 2$  has a cycle.  
 (e) [3 marks] Prove that every graph  $G = (V, E)$  where  $\forall v \in V, d(v) \geq |V| - 3$  and  $|V| > 4$  is connected.  
**Hint:** Assume the graph is not connected and count the vertices adjacent to a pair of disconnected vertices.  
 (f) [3 marks] Add the degrees in the previous part and use this sum to calculate the number of edges. Compare your result to Example 6.6 and Example 6.7 from the course notes and explain the connection (or lack thereof).

2. [6 marks] **binary representation** Read Theorem 4.1, as well as its proof. It is useful to insist that there be a unique (one and only one) binary representation for integers. This can be done by specifying the number of bits in the representation:

- (a) [3 marks] Prove the following:

For every number  $n \in \mathbb{N}^+$ , there is a unique representation of  $n$  in the following form:  $n = \sum_{i=0}^p b_i 2^i$ , where  $p$  is the smallest integer such that  $2^{p+1} > n$ ,  $p$  is non-negative, and  $b_p, \dots, b_0 \in \{0, 1\}$ .

- (b) [3 marks] Use the previous question and some reasoning about 0 to show that there is a unique binary representation of every natural number. Explain why it wasn't just possible to make the domain of the previous proof "every number  $n \in \mathbb{N}$ "?

3. [14 marks] **algorithm analysis**

- [3 marks] Prove the claim in Exercise 5.4. The useful formula should be a summation over  $ir^i$  rather than  $ix^i$ .
- [2 marks] Use precise reasoning to find the average run-time in Exercise 5.4 if the words "1 and 10" are replaced by "1 and 500."
- [3 marks] Define the input size of counter (below) as  $n = s + u$ . Find, and prove, a big-Theta bound in terms of  $n$  for the worst-case run-time of counter.
- [3 marks] Now, find and prove, a big-Theta bound in terms of  $n$  for the best-case run-time of counter.

- [3 marks] Based on the previous two questions, what (if anything) can you say about a big-Theta bound on the average run-time of counter? Explain.

```
1  # assume s, u are natural numbers
2  # and that u < 7
3  def counter(s, u):
4      steps = 0
5      while s + u > 0:
6          if u == 0:
7              u = 6
8              s = s - 1
9          else:
10             u = u - 1
11             steps = steps + 1
12     return steps
```