**CSCI 2200 — Foundations of Computer Science (FoCS)**
**Homework 4 (document version 1.1)**

## Overview

- This homework is due by 11:59PM on Thursday, November 2
- You may work on this homework in a group of up to four students; unlike recitation problem sets, **your teammates may be in any section**
- You may use at most **two** late days on this assignment
- Please start this homework early and ask questions during office hours and at your November 1 recitation section; also ask questions on the Discussion Forum
- Please be concise in your written answers; even if your solution is correct, if it is not well-presented, you may still lose points
- You can type or hand-write (or both) your solutions to the required graded problems below; **all work must be organized in one PDF that lists all teammate names**
- You are strongly encouraged to use LaTeX, in particular for mathematical symbols; see the corresponding `hw1.tex` file as a starting point and example

## Grading

- For each assigned problem, a grade of 0, 1, or 2 is assigned as follows: 0 indicates no credit; 1 indicates half credit; and 2 indicates full credit
- No credit is assigned if a problem is not attempted or minimal work/progress is shown
- Half credit is assigned if a strong attempt was made toward a solution and/or only part of the problem was attempted or solved
- Full credit is assigned for a perfect or nearly perfect solution, i.e., only one or two minor typos/mistakes at most

## Warm-up exercises

The problems below are good practice problems to work on. Do not submit these as part of your homework submission. **These are ungraded problems.**

- **Problem 8.12(a-c).**
- **Problem 9.8(a).**
- **Problem 9.20.**
- **Problem 9.23(a).**
- **Problem 9.28.**
- **Problem 9.43.**
- **Problem 9.69-9.72.**
- **Problem 11.3.**
- **Problem 11.10.**
- **Problem 11.13.**
- **Problem 11.15(a,c-d).**

## Graded problems

The problems below are required and will be graded.

- ***Problem 9.23(b).**
- ***Problem 9.37. (v1.1) See correction below.**
- ***Problem 9.39.**
- ***Problems 11.5 and 11.15(b).**
- ***Problem 11.11.**
- ***Problem 11.17.**
- ***Problem 11.27.**

As you might not have the required textbook yet, all of the above problems (both graded and ungraded) are transcribed in the pages that follow.

Graded problems are noted with an asterisk (*).

If any typos exist below, please use the textbook description.

- **Problem 8.12(a-c).** A set $\mathcal{P}$ of parenthesis strings has a recursive definition (right).

  (1) $\varepsilon \in \mathcal{P}$

  (2) $x \in \mathcal{P} \rightarrow [x] \in \mathcal{P}$
  $x, y \in \mathcal{P} \rightarrow xy \in \mathcal{P}$

  (a) Determine if each string is in $\mathcal{P}$ and give a derivation if it is in $\mathcal{P}$.
  (i) [[[]]][ (ii) [][[]][[]] (iii) [[][][]

  (b) Give two derivations of [][][[]] whose steps are not a simple reordering of each other.

  (c) Prove by structural induction that every string in $\mathcal{P}$ has even length.

- **Problem 9.8(a).** Estimate these sums.

  (a) $\displaystyle\sum_{i=1}^{10}\sum_{j=1}^{20} 2^{i+j}$

- **Problem 9.20.** Prove or disprove:

  (a) $\dfrac{n^3+2n}{n^2+1} \in \Theta(n)$

  (b) $(n+1)! \in \Theta(n!)$

  (c) $n^{1/n} \in \Theta(1)$

  (d) $(n!)^{1/n} \in \Theta(n)$

- **Problem 9.23(a).** Prove by contradiction: (a) $n^3 \notin O(n^2)$

- ***Problem 9.23(b).** Prove by contradiction: (b) $2^n \notin \Theta(3^n)$

Let us assume $2^n \in \Theta(3^n)$

By definition, if a function is in $\Theta(3^n)$, it means that it is both in $\Omega(3^n)$(Lower Bound) and $O(3^n)$(Upper Bound)

From this, we can define $2^n \in \Omega(3^n)$ and $2^n \in O(3^n)$

From the lower bound ($\Omega(3^n)$), there exists a $n_0 \in \mathbb{N}$ and a constant c $> 0$ such that

$c * 3^n \leq 2^n \ \forall \ n \geq n_0$, now we can simplify

$c \leq \frac{2^n}{3^n} \ \forall \ n \geq 0$

to get $c \leq (\frac{2}{3})^n \ \forall \ n \geq 0$

but for all $c > 0$, we see that there exists a $k_0$ such that $(\frac{2}{3})^k \leq c \ \forall \ k \geq k_0$, which contradicts the inequality above

From this contradiction, we see that $2^n \notin \Omega(3^n)$

Therefore, because the function is not in the lower bound of $\Omega(3^n)$, we have proven that $2^n \in \Theta(3^n)$ is false, in turn proving that $2^n \notin \Theta(3^n)$ is true

- **Problem 9.28.** For recurrence $f(0) = 1$; $f(n) = nf(n-1)$, compare $f(n)$ with (a) $2^n$ (b) $n^n$.

- ***Problem 9.37.** A recursive algorithm has a runtime $T(n)$ that depends only on $n$, the input of size. $T(1) = 1$ and for an input-size $n$, the algorithm solves two problems of size $\lfloor n/2 \rfloor$ and does extra work of $n$ to get the output. (**(v1.1)** Note "ceiling" was corrected to "floor" here.)

  (a) Argue that $T(n)$ satisfies the recursion $T(n) = 2T(\lfloor n/2 \rfloor) + n$.

  I argue that $T(n)$ satisfies the recursion $T(n) = 2T(\lfloor n/2 \rfloor)+n$ because it is clearly stated that a recursive algorithm has a runtime of T(n) that *only* depends on n, the input of size. It is then said that for the input-size $n$, the algorithm solves two problems of size $\lfloor n/2 \rfloor$, which is just $2T(\lfloor n/2 \rfloor)$ and does extra work for n, which is just $+\ n$. Putting it all together, it comes out as the recursion $T(n) = 2T(\lfloor n/2 \rfloor) + n$

  (b) Prove $T(n) \in \Theta(n \log n)$.
  [Hint: Induction to show $n \log_2 n \leq T(n) \leq 2n \log_2 n$ for $n = 2^k$ and monotonicity.]

  First of all, unfold T(n)
  $T(n) = 2T(\lfloor n/2 \rfloor) + n$
  $2T(\lfloor n/2 \rfloor) = 4T(\lfloor n/4 \rfloor) + n$
  $4T(\lfloor n/4 \rfloor) = 8T(\lfloor n/8 \rfloor) + n$
  ...
  $2^{n-1}T(...) = 2^k T(1) + n * k$
  Then you get

  T(n) = $2^k + n * k$
  Then we can conclude that if $2^k = n$, then $\log_2(n)$ = k, plugging it in, we get
  $T(n) = n \log_2(n) + n$
  Now we want to prove T(n) $\rightarrow$ T(n+1) for n $\geq$ 1 by induction
  [Base Case] T(1) = $(1) \log(1) + 1 = 1 = 1$ Which is true
  [Induction Step] We show T(n) $\rightarrow$ T(n+1) for all n $\geq$ 1
  Assume (induction hythesis) that T(n) is true
  We want to prove T(n+1) = $(n + 1) \log_2(n + 1) + (n + 1)$
  LHS:
  $T(n + 1) = 2T(\lfloor \frac{n+1}{2} \rfloor) + n + 1$
  $= 2((\frac{n+1}{2}) \log_2(\frac{n+1}{2}) + \frac{n+1}{2}) + n + 1$
  $= (n + 1) \log_2(\frac{n+1}{2}) + 2n + 2$
  $= (n + 1)(\log_2(n + 1) - 1) + 2n + 2$
  $= \log_2(n + 1) + n \log_2(n + 1) - (n + 1) + 2n + 2$
  $= \log_2(n + 1) + n \log_2(n + 1) + n + 1$
  $= (n + 1)(\log_2(n + 1)) + n + 1$
  Thus proving that T(n+1) = $(n + 1) \log_2(n + 1) + (n + 1)$
  Then taking the limit of n to inf

  $$\lim_{n\to\infty} \frac{n \log_2(n)}{n \log_2(n)} + \frac{n}{n \log_2(n)} = 1$$

  Since we get a constant, we have proved that T(n) $\in \Theta(n \log(n))$

- ***Problem 9.39.** Use the rule of thumb for nested sums on the bottom of page 118 (i.e., You can quickly determine the growth rate of a nested sum using: growth rate of nested sum = number of nestings + order of the summand) to obtain the asymptotic growth rate for the following sums, and verify by exact computation. If the rule does not work, why not?

  (a) $\sum_{i=1}^{n}\sum_{j=1}^{i} j$

  (b) $\sum_{i=1}^{n}(i^2 + \sum_{j=1}^{n} j)$

  (c) $\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(i^2 + ijk)$

  (d) $\sum_{i=1}^{n}\sum_{j=1}^{i^2} j$

- **Problem 11.3.** Give the degree sequences of $K_{n+1}$, $K_{n,n}$, $L_n$, $C_n$, $S_{n+1}$, and $W_{n+1}$.

- ***Problems 11.5 and 11.15(b).** A graph is regular if every vertex has the same degree. Which of these graphs are regular?

  (a) $K_6$ (b) $K_{4,5}$ (c) $K_{5,5}$ (d) $L_6$ (e) $S_6$ (f) $W_4$ (g) $W_5$

  A graph is $r$-regular if every vertex has the same degree $r$. Show:

  (b) If $r$ is odd and $n$ is odd, there is no $r$-regular graph with $n$ vertices.

- **Problem 11.10.** Give graphs with these degree distributions, or explain why you can't. Verify $2|E| = \sum_{i=1}^{n} \delta_i$.

  (a) $[5, 3, 3, 2, 1]$ (d) $[3, 3, 3, 3, 3]$ (g) $[4, 4, 4, 4, 4]$ (j) $[3, 3, 3, 2, 2]$
  (b) $[3, 2, 1, 1, 1]$ (e) $[3, 3, 3, 3, 3, 3]$ (h) $[4, 4, 3, 2, 1]$ (k) $[3, 3, 3, 3, 2]$
  (c) $[3, 3, 2, 1]$ (f) $[3, 3, 2, 2, 2]$ (i) $[4, 3, 3, 2, 2]$ (l) $[5, 3, 2, 2, 2]$

- ***Problem 11.11.** In a graph only the two vertices $u, v$ have odd degree. Prove there is a path from $u$ to $v$.

- **Problem 11.13.** Compute the number of edges in the following graphs:

  (a) $K_n$ (b) $K_{n,\ell}$ (c) $W_n$

- **Problem 11.15(a,c-d).** A graph is $r$-regular if every vertex has the same degree $r$. Show:
  (a) If $r$ is even and $n > r$, there is an $r$-regular graph with $n$ vertices. (Tinker!)
  (c) If $r$ is odd and $n > r$ is even, there is an $r$-regular graph with $n$ vertices.
  (d) An $r$-regular graph with $4k$ vertices must have an even number of edges.

- ***Problem 11.17.** A graph $G$ has $n$ vertices.
  (a) What is the maximum number of edges $G$ can have and not be connected? Prove it.
  (b) What is the minimum number of edges $G$ can have and be connected? Prove it.

- ***Problem 11.27.** Every vertex degree in a graph is at least 2. Prove that there is at least one cycle.