CSc 30400 Introduction to Theory of Computer Science

5th Homework Set

Due Date: 5/11

Instructions:

- This homework set covers Chapter 7 (Complexity).
- Submit your solutions by email or hand them in class **before the beginning of the class!**
- Exercises are divided into two sections: Easy and Hard. "Easy questions" should be relatively easy. "Hard questions" require more understanding and effort than the easy ones.
- Every question gains 10 points unless otherwise stated. The maximum amount of points you can get is 80.

Academic Integrity:

- You are allowed to use *trustworthy* sources from the internet in order to prepare your homework. If you do so you should always cite the source you are using in your report. You are not allowed to copy material from the internet, you should always give your own interpretation of the cited material.
- You are allowed (and encouraged) to work in groups of 2 or 3 people for this homework. If you decide to do so you should state this clearly on your report. Every student should submit his/her own report. You should not copy work from your colleagues' reports!

1 Easy questions

E 1. We said that o, ω, Θ resemble $<, >, =$. Place the following functions in a table in decreasing order: if $f = o(g)$ then $g(n)$ will be placed above $f(n)$ in the table; functions having a Θ relation ($f = \Theta(g)$) should be placed in the same level.

Example: $5n^2 + 2n$, $10n$, n^2

$5n^2 + 2n$, n^2
$10n$

Functions: $2n^2+5n-4$, $2^{10}n$, $3n^3+5n+4$, n, $\frac{1}{5}n^2$, 2^n, $\log_2 n$, $n!$, $\log_{10} n$.

E 2. Compare the following functions (justify your answer). Use O-notation (o, ω, Θ).

Example:

- $n^2, 5n^2 + 2n$
  $n^2 = \Theta(5n^2 + 2n)$
- $10n, n^2$
  $10n = o(n^2)$

(a) $2^{(n+1)}$, 2^n

(b) 2^n, 3^n

(c) $\log_5 n^2$, $\log_2 n$

(d) $\log_2(n!)$, n^2

Hint: Use the property of logarithms regarding multiplication: $\log xy = \log x + \log y$.

E 3. (15 points) In the following algorithms you have to determine the number of assignments that are performed regarding the variable `x`. You don't have to be exact (use the O-notation).

Example:

```
x=1;
for (i=1; i<=n; i++)
    x=x+1;
```

This algorithm performs n iterations and inside each iteration a constant number of assignments, so it performs in $O(n)$ assignments.

(a)
```
x=1;
for (i=n; i>1; i--)
    x=x+1;
```

(b)
```
x=1;
for (i=1; i<=n; i+=2)
    x=x+1;
```

(c)
```
x=1;
for (i=1; i<=n; i++) {
    for (j=1; j<=n; j++)
        x=x+1;
}
```

(d)
```
x=1;
for (i=1; i<=n; i++) {
    for (j=1; j<=i; j++)
        x=x+1;
}
```

(e)
```
x=1;
for (i=n; i!=1; i=i/2)
    x=x+1;
```

E 4. (20 points) Assume that we are given two decision problems A and B. Here are some statements regarding these two problems:

a. A is in P
b. A is in NP
c. A is NP-complete
d. B is in P
e. B is in NP
f. B is NP-complete
g. A $\leq_P$ B
h. B $\leq_P$ A

Answer the following questions and justify your answers.

Example: Does (a) imply (b)?

Yes! The class of problems in P is a subset of the class NP, so every problem in P is also in NP.

i. Does (b) imply (a)?

ii. Does (b) imply (c)?

iii. Is it possible that both (a) and (c) hold?

iv. Do (a) and (h) imply (d)?

v. Do (b) and (g) imply (e)?

vi. Do (b) and (g) imply (f)?

vii. Do (c) and (g) imply (f)?

viii. Do (a) and (f) imply (g)?

ix. Do (c) and (f) imply (g)?

x. Does (g) imply (h)?

2 Hard questions

H 1. Prove the following two statements:

(a) If $f = O(g)$ then $g = \Omega(f)$

(b) If $f = O(g)$ and $f = \Omega(g)$ then $f = \Theta(g)$

H 2. (5 points) Suppose that we are given a number n in binary as input. The goal is to construct an algorithm that transforms this number to unary representation. Is this problem in P?

H 3. The Longest Path Problem is the following: given a graph G and an integer k determine whether G has a path of length at least k that does not visit any vertex twice.

(a) Show that the Longest Path Problem is in NP.

(b) Give a polynomial time reduction from the Hamilton Path problem to the Longest Path problem.

(c) Does there exist a polynomial-time algorithm for the Longest Path problem? Justify your claim.