Homework 1

Strukture podataka i algoritmi I - I053

Homework instructions

The submission deadline is October 18, 2023 at 9:00 (just before the exercise session). You can type the tasks in LaTeX or write them by hand and scan them. Programming tasks should be submitted as .cpp files. All files need to be submitted to Teams. You can achieve a maximum of 100 points.

Task 1 (5+5+5+5 pts.)**.** Use the definiton of asymptotic notation (i.e. find c and n_0) to prove the following statements:

a) $\log_2(n) = \Theta(\ln n)$

b) $3n\sqrt{n} + 6n \ln n - 4n = O(n^2)$

c) $3n\sqrt{n} + 6n \ln n - 4n = O(n^3)$

d) $\sum_{i=0}^{\lfloor \log_4 n \rfloor} 4^i = \Theta(n)$

Task 2 (10 pts.)**.** Explain how would you interpret the following expression:

$$8n^2 + 5n + \Theta(n \log n) = \Theta(n^2).$$

Task 3 (20 pts.)**.** The asymptotic bound provided by O-notation is not tight. For example:

- $5n^2 = O(n^2)$ (tight),
- $5n^2 = O(n^{10})$ (not tight).

That's why we introduce o-notation to denote a non-asymptotically-tight upper bound. Formally, the definition is

$$o(g(n)) = \{f(n) : \text{ for any } c > 0 \text{ there exists } n_0 > 0 \text{ such that } 0 \leq f(n) < cg(n) \text{ for all } n \geq n_0\}.$$

Compared to O-notation, where we required $0 \leq f(n) \leq cg(n)$ to hold for **some** c and **some** n_0, now we require it to hold for **all** positive constants c. Note that n_0 can be different for different constants c. Your task is to prove $n! = o(n^n)$.

Task 4 (10+10+20+10 pts.)**.**

a) Write a C++ program that implements the Selection Sort algorithm. Test it on some 5-element array.

b) Give a tight upper bound on the worst-case running time $T(n)$. Prove it by mathematical induction.

c) Prove your algorithm is correct.

d) Write a C++ program that generates 100 random 1000-element arrays of integers, time the selection sort algorithm on each array and report the average running time. What happens if you increase the array size to 1000000?