Data structure and algorithms
(2024/25)

Homework assignment 1 - theoretical part

Problem 1:

(i) Rank the following functions by asymptotic growth rate in non-decreasing order:

$$f_1(n) = 2^{2^{1000000}}, f_2(n) = 2^{1000000n}, f_3(n) = \binom{n}{2}, f_4(n) = n\sqrt{n}.$$

Justify your answer!

Example: The function $f(n) = n$ grows asymptotically slower then the function $g(n) = n^2$; that is, $f(n) \in \mathcal{O}(g(n))$, but $g(n) \notin \mathcal{O}(f(n))$.

(ii) Using big $\mathcal{O}$-notation show that $n^{1+0{,}001} \notin \mathcal{O}(n)$.

Problem 2: Let f and g be two positive functions. Are the following propositions true or false? Justify your answer.

(i) $f(n) = O(g(n)) \Leftrightarrow 2^{f(n)} = \Omega(2^{g(n)})$;

(ii) $f(n) = \Theta(f(\frac{n}{2}))$;

(iii) $f(n) + g(n) = \theta(\min\{f(n), g(n)\})$, where

$$\min\{f(n), g(n)\}(n) = \begin{cases} f(n); & \text{if } f(n) \leq g(n) \\ g(n); & \text{otherwise.} \end{cases}$$

Problem 3: Consider the searching problem:

Input: A sequence of n numbers $A = [a_1, a_2, \ldots, a_n]$ and a value v.
Output: An index i such that $v = A[i]$ or the special value NILL if v does not appear in A.

(i) Write pseudocode for linear search, which scans through the sequence, looking for v.

(ii) Using a loop invariant, prove that your algorithm is correct.

Problem 4: Let A be an integer array of length n. The following algorithm is given:

$foo(A)$
  $n \leftarrow A.length$
  $x \leftarrow A[1]$
  for $i \leftarrow 2$ **to** n **do**
    if $A[i] > x$ **then**
      $x = A[i]$
  return x

(i) What does $foo(A)$ return?

(ii) Write a loop invariant for $foo(A)$.

(iii) Using a loop invariant, prove that $foo(A)$ is correct.

Problem 5: The following algorithm is given:

$int\ foo(int\ x, y)$
  if $x == 0$ **return** y
  $y = 2 \cdot y + x\%2$
  return $foo(x/2, y)$

(i) Compute $foo(5,\ 0)$, $foo(6,\ 0)$, $foo(7,\ 0)$.

(ii) What is the time complexity of the call $foo(n,\ 0)$?

(iii) Let $f_1(n) = 12n + 11\log(n)$ and $f_2(n) = \frac{11}{n} + 12\log(n)$. For each function, find to which of the following families it belongs:

$$O(n), \Omega(n), \Theta(n), O(\log n), \Omega(\log n), \Theta(\log n).$$

Problem 6: The following algorithm is given:

$int\ FooBar(c, n)$:
  $int\ p = 2 \star\star n$
  result $= 1$
  while $(p > 0)$:
    result $=$ result $\star c$
    $p = p - 1$
  return result

(i) What is the time complexity of the function *FooBar* in terms of n? Justify your answer.

(ii) Can you speed up the function *FooBar*? If yes, how and what is the time complexity now, and if not, why not.

(iii) Prove that:

$$n \ln n = O(n^{3/2})$$

and

$$n \ln n = \Omega(n^{1/2}).$$

The solutions of the problems should be submitted via e-classroom: **only one .pdf file should be uploaded**. The deadline for the submission is **Sunday, November 10**, 2024.