Solutions for Data Structures and Algorithms Spring 2023 — Problem Sets

By Dmitriy Okoneshnikov, B22-DSAI-04

January 23-24, 2023

Week 1. Problem set

1. Compute asymptotic worst case time complexity of the following algorithm (see pseudocode conventions in [Cormen, Section 2.1]). You **must** use Θ-notation. For justification, provide execution cost and frequency count for each line in the body of the secret procedure. Optionally, you may provide details for the computation of the running time $T(n)$ for worst case scenario. Proof for the asymptotic bound is not required for this exercise.

```
/* A is a 0-indexed array,
 * n is the number of items in A */
secret(A, n):
k := 0
for i = 1 to n-1
    k := k + 1
    j := i
    while j < n and A[j-1] ≥ A[j]
        j := j + k
    exchange A[i] with A[j]
```

Solution.

pseudocode	*cost*	*times*
`/* A is a 0-indexed array,` `* n is the number of items in A */`	0	1
`secret(A, n):`	0	1
`k := 0`	c_1	1
`for i = 1 to n-1`	c_2	n
`k := k + 1`	c_3	$n-1$
`j := i`	c_4	$n-1$
`while j < n and A[j-1] ≥ A[j]`	c_5	$\sum_{i=1}^{n-1} t_i$
`j := j + k`	c_6	$\sum_{i=1}^{n-1} (t_i - 1)$
`exchange A[i] with A[j]`	c_7	$n-1$

Therefore, worst case scenario: $T(n) = 0\cdot 1 + 0\cdot 1 + c_1\cdot 1 + c_2\cdot n + c_3\cdot(n-1) + c_4\cdot(n-1) + c_5\cdot\sum_{i=1}^{n-1} t_i + c_6\cdot\sum_{i=1}^{n-1}(t_i-1) + c_7\cdot(n-1) = c_1\cdot 1 + c_2\cdot n + c_3\cdot(n-1) + c_4\cdot(n-1) + c_5\cdot\frac{n\cdot(n-1)}{2} + c_6\cdot\frac{(n-1)\cdot(n-2)}{2} + c_7\cdot(n-1) = c_1 + c_2\cdot n + c_3\cdot n - c_3 + c_4\cdot n - c_4 + \frac{c_5}{2}\cdot n^2 - \frac{c_5}{2}\cdot n + \frac{c_6}{2}\cdot n^2 - \frac{3c_6}{2}\cdot n + c_6 + c_7\cdot n - c_7 = (\frac{c_5}{2} + \frac{c_6}{2})\cdot n^2 + (c_2 + c_3 + c_4 - \frac{c_5}{2} - \frac{3c_6}{2} + c_7)\cdot n + (c_1 - c_3 - c_4 + c_6 - c_7) = \Theta(n^2)$

2. Indicate, for each pair of expressions (A, B) in the table below whether $A = O(B)$, $A = o(B)$, $A = \Omega(B)$, $A = \omega(B)$, or $A = \Theta(B)$. Write your answer in the form of the table with *yes* or *no* written in each box:

A	B	$A = O(B)$	$A = o(B)$	$A = \Omega(B)$	$A = \omega(B)$	$A = \Theta(B)$
$\log^5 n$	$\sqrt[4]{n}$					
n^{1000}	1.0001^n					
$n^{\cos n}$	$\log n$					
3^n	$3^{0.5n}$					

Solution.

A	B	$A = O(B)$	$A = o(B)$	$A = \Omega(B)$	$A = \omega(B)$	$A = \Theta(B)$
$\log^5 n$	$\sqrt[4]{n}$	yes	yes	no	no	no
n^{1000}	1.0001^n	yes	yes	no	no	no
$n^{\cos n}$	$\log n$	no	no	no	no	no
3^n	$3^{0.5n}$	no	no	yes	yes	no

3. Let f and g be functions from positive integers to positive reals. Assume $f(n) > n$ for $n > 0$. Using definition of asymptotic notation, prove formally that

$$\min(f(n) - n, g(n) + n) = O(f(n) + g(n))$$

Solution.

Proof. We need to show that there $\exists$ constants c and n_0, such that $\forall n \geq n_0$ we have $\min(f(n) - n, g(n) + n) \leq c \cdot (f(n) + g(n))$.

Let $c = 1, n_0 = 1$ Consider two cases:

(a) $\min(f(n) - n, g(n) + n) = f(n) - n$. Then, we have $f(n) - n = O(f(n) + g(n))$.
$f(n) - n \leq c \cdot (f(n) + g(n))$
$f(n) - n \leq (f(n) + g(n))$
$0 \leq g(n) + n, g(n) > 0, n > 0 \Rightarrow$ it is true

(b) $\min(f(n) - n, g(n) + n) = g(n) + n$. Then, we have $g(n) + n = O(f(n) + g(n))$.
$g(n) + n \leq c \cdot (f(n) + g(n))$
$g(n) + n \leq (f(n) + g(n))$
$0 \leq f(n) - n, f(n) > n, n > 0 \Rightarrow$ it is true

Thus, in both cases we have shown the inequality holds for all $n \geq n_0$.

QED.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms, Fourth Edition.* The MIT Press 2022