DSA. Problem solutions. Week 1.

Maksim Al Dandan

April 3, 2024

1 Task 1

1.1 Statement

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 the 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 := 2 * j
    exchange A[i] with A[min(j, n - 1)]
```

1.2 Solution

Cost	Times
c_4	1
c_5	n
c_6	$n-1$
c_7	$n-1$
c_8	$n(n-1)$
c_9	$(n-1)^2$
c_{10}	$n-1$

$$T(n) = c_4 * 1 + c_5 * n + c_6 * (n-1) + c_7 * (n-1) + c_8 * n(n-1) + c_9 * (n-1)^2 + c_{10} * (n-1) \quad (1)$$

$$T(n) = c_4 + nc_5 + c_6(n-1) + c_7(n-1) + nc_8(n-1) + c_9(n-1)^2 + c_{10}(n-1) \quad (2)$$

Answer: $T(n) = \Theta(n^2)$

2 Task 2

2.1 Statement

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.

2.2 Answer

A	B	$A = O(B)$	$A = o(B)$	$A = \Omega(B)$	$A = \omega(B)$	$A = \Theta(B)$
1.0001^n	n^{1000}	yes	yes	no	no	no
3	$(1+1/n)^n$	yes	yes	no	no	no
$n^{\sin n}$	$\log_2 n$	no	no	no	no	no
$\log_2^3 n$	$\sqrt[6]{n}$	no	no	yes	yes	no

3 Task 3

3.1 Statement

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

$$max(f(n) + \sqrt{n}, g(n) - n) = O(f(n) + g(n))$$

3.2 Solution

Proof:

We need to show that there exist constants c and n_0, such that for all n $\geq n_0$ we have.
Let us consider two cases:

1. $max = f(n) + \sqrt{n}$
It implies that: $f(n) + \sqrt{n} \leq f(n) + g(n)$, where g(n) $>$ n
Thus, $f(n) + \sqrt{n} \leq f(n) + n$ (at least)
But n grows faster than $\sqrt{n}$, hence it works.

2. $max = g(n) - n$
It implies that: $g(n) - n \leq f(n) + g(n)$, where g(n) $>$ n
Thus, $f(n) \geq -n$, but f(n) due to condition contains only of positive reals
Hence, it works always.

QED