Daniil Manakovskiy
BS18-02

Assignment 1

Theoretical part

1 Problem A

The worst case for given algorithm is the situation when B contains all elements of A

```
for i ← 0 to n − 1 ;                              // c_1 · (n + 1)
do
    contained ← FALSE ;                           // c_2 · n
    for j ← 0 to n − 1 ;                          // c_3 · n · (n + 1)
    do
        if A[i] = B[i];                           // c_4 · n^2
        then
            contained ← True ;                    // c_5 · n^2
    end
    if not contained ;                            // c_6 · n
    then
        return FALSE ;                            // c_7 · 0
end
return TRUE ;                                     // c_8 · 1
```

Algorithm 1: Given algorithm with execution time of every step in the worse-case

1.1 Running time calculation

$T(n)$ is a function of algorithm's running time based on the input size n

$$\begin{aligned} T(n) = c_1 \cdot (n+1) + c_2 \cdot n + c_3 \cdot n \cdot (n+1) + c_4 \cdot n^2 + c_5 \cdot n^2 + c_6 \cdot n + c_7 \cdot 0 + c_8 \cdot 1 = \\ = (c_3 + c_4 + c_5) \cdot n^2 + (c_1 + c_2 + c_3 + c_6) \cdot n + c_1 + c_8 \end{aligned}$$

Let

$$c_3 + c_4 + c_5 = A, c_1 + c_2 + c_3 + c_6 = B, c_1 + c_8 = C$$

then

$$T(n) = An^2 + Bn + C$$

Theorem 1.1. $T(n) = O(n^2)$

Proof.

$$T(n) = O(g(n))\colon\ \exists\, c_1 > 0, n_0 > 0\colon\ 0 \le T(n) \le c_1 \cdot g(n)\, \forall\, n \ge n_0$$

If

$$c_1 = 2A, g(n) = n^2$$

then

$$0 \le An^2 + Bn + C \le 2An^2\, \forall\, n \ge n_0$$

$$-An^2 + Bn + C \leq 0$$

Since A, B, C are all positive

$$D = B^2 + 4AC > 0$$

$$n_0 = \frac{b^2 + \sqrt{D}}{2A}$$

Therefore $c_1 = 2A$ and $n_0 = \frac{b^2+\sqrt{D}}{2A}$ are satisfies to requirements to $T(n) = O(n^2)$ □

2 Problem B

Theorem 2.1. *For arrays $A = [1, 1, 1, ..., 1]$ and B such that $1 \in B$, $|B| = |A|$ time complexity of given algorithm is $\Omega(n^2)$*

Proof. Indeed, for given arrays the algorithm behaves like in the worst case described in 1.1. Therefore it's running time $T(n) = An^2 + Bn + C$

$$T(n) = \Omega(g(n)) : \exists\, c_1 > 0, n_0 > 0 : 0 \leq c_1 \cdot g(n) \leq T(n) \,\forall\, n \geq n_0$$

If

$$c_1 = A, g(n) = n^2$$

then

$$An^2 \leq An^2 + Bn + C \,\forall\, n \geq n_0$$

$$0 \leq Bn + C$$

Since B, C are all positive, resulting inequality is always true, therefore

$$T(n) = \Omega(n^2)$$

□

3 Problem C

Yes, the worst-case of the algorithm is $\Theta(n^2)$

Theorem 3.1. $T(n) = \Theta(n^2)$

Proof. Since

$$T(n) = \Theta(n^2) \iff T(n) = O(n^2) \quad \&\& \quad T(n) = \Omega(n^2)$$

Both equalities were proven in 1.1 and 2.1. □