Kacper Kalinowski 01.02.21 r.

W trakcie kolokwium ASD nr 2 nie korzystałem z żadnej pomocy innych osób.

Kalinowski 01.02.21 r.

1.

function Member (n, A[1..n], x)

i = 1 — 1 1

found = false — 1 1

while (i <= n) and (not found) do — 1

if A[i] < x

then i = i + 1 — 1

else found = true

return found

1 a) 2 + n

1 b) = 2

2. Q , A , S

function A(Q [1..n], n)

i = 1

while Queue_Empty (Q) == false do

A[i] = Dequeue (Q)

i = i + 1

x = 1 , j = 1

for x to n

for j to n

if A[j] < A[j+1] then

swap (A[j], A[j+1])

Kacper Kalinowski 01.02.21 r.

2 cd.

```
x = 1
for x to n
    Push (S, A[x])

Return S
```

3.

L

~~Procedure~~

```
function B (L, L.tail)
    x = L.head
    i = 1, A[1..n]
    while x.next <> nil L do
        if x mod 2 == 1 then
            A[i] = x
            i = i + 1
        x = x.next
    suma = 1
    x = 1
    for x to i
        suma = suma * A[x]

    return sqrt(i, suma)
```

Yaşar Kalınaslı 01.02.2021

4. $\bar{T}$ $a_1^{w_1} + a_2^{w_2}$ a_1 - min, a_2 maks
w_1 - liczba wierzchołków

~~Function G(T)~~

```
Function minT(T)
    p=T.root
    min=p
    if p<>NIL
        min=minT(p.left)
    return min
Function maxT(T)
    p=T.root
    max=p
    if p<>NIL
        max=maxT(p.right)
    return max

Function MinK(T)
    i=1
    x=T.root
    while x.left<>NIL do
        x=x.left
        i=i+1
    return i
Function MaxK(T)
    i=1
    x=T.root
    while x.right<>NIL do
        x=x.right
        i=i+1
    return i
```

Kaya Kolinski 01.02.21

4 cd.

Function C (T)

$a_x = \min I(T)$

$a_y = \max T(T)$

$\omega_x = \min K(T)$
$\omega_y = \max K(T)$

$sumax = pwr(a_x, \omega_x)$
$sumay = pwr(a_y, \omega_y)$

return sumax + sumay

Kacper Kalinowski 01.02.2021

5. A = [3, 7, 15, 21, 22, 20, 19, 13, 14, 21]

1. 1:3

2. 1:3
2:7

3. 1:3
2:7 3:15

4. 1:3
2:7 3:15
4:21

5. 1:3
2:7 3:15
4:21 5:22

6. 1:3
2:7 3:15
4:21 5:22 6:20

7. 1:3
2:7 3:15
4:21 5:22 6:20 7:19

8. 1:3
2:7 3:15
4:13 5:22 6:20 7:19
8:21

9. 1:3
2:7 3:15
4:13 5:22 6:20 7:19
8:21 9:14

10. 1:3
2:7 3:15
4:13 5:21 6:20 7:19
8:21 9:14 10:22

6.

preorder: [3, 7, 13, 21, 14, 21, 22, 15, 20, 19]

inorder [21, 13, 14, 7, 22, 21, 3, 20, 15, 19]

postorder [21, 14, 13, 22, 21, 7, 20, 19, 15, 3]