Example 2 ($n \in \mathbb{N}^+$)

```
for i ← 1 to n
    for j ← 1 to i
        print „Hello"   θ1
print „Bye"   θ1
```

$$\Theta(1)+\sum_{i=1}^{n}\sum_{j=1}^{i}\Theta(1) =$$

$$= \Theta(1)+\sum_{i=1}^{n}\Theta(i) = \Theta(1)+\frac{n.(n+1)}{2} = \Theta(1)+\Theta(n^2) = \Theta(n^2)$$

Example 3 ($n \in \mathbb{N}^+$)

```
a ← 0
for i ← 1 to n
    for j ← i+1 to n
        for k ← 1 to j
            a ← a+1   θ(1)
return a   θ(1)
```

$$1+\sum_{i=1}^{n}\sum_{j=i+1}^{n}\sum_{k=1}^{j} 1 = \sum_{i=1}^{n}\sum_{j=i+1}^{n} j =$$

$$= \sum_{i=1}^{n}\left(\sum_{j=1}^{n} j - \sum_{j=1}^{i} j\right) = \sum_{i=1}^{n}\left(\frac{n.(n+1)}{2} - \frac{i.(i+1)}{2}\right) =$$

$$= \sum_{i=1}^{n}\frac{n(n+1)}{2} - \sum_{i=1}^{n}\frac{i(i+1)}{2} = \frac{n.n(n+1)}{2} - \frac{1}{2}\sum_{i=1}^{n} i^2 - \frac{1}{2}\sum_{i=1}^{n} i =$$

$$= \frac{n.n(n+1)}{2} - \frac{1}{2}\left(\frac{n(n+1)(2n+1)}{6}\right) - \cancel{\frac{1}{2}\,\frac{n(n+1)}{2}} =$$

$$= \frac{n^3+n^2}{2} - \frac{1}{6}n^3 = \Theta(n^3)$$

Доказателства за коректност

15.03.24

задача MYSTERY (A[1...n] масив от числа)

```
1 for i ← 1 to n
2     for j ← 1 to n-i
3         if A[j] > A[j+1]
4             swap(A[j], A[j+1])
```

4	3	6	8	1
3	4	6	8	1
3	4	6	8	1
3	4	6	8	1
3	4	6	1	8

Инвариант 1: при всяко достигане на ред 1 A[n+2-i...n] съдържа i-1 най-големи елементи на входа в сортиран (възх.) ред

Инвариант 2: при всяко достигане на ред 2 i-тият по големина елем. на входа е max елем. на подмасива A[j...n+1-i]

Допускаме, че инв. 2 е изпълнен. Док. инв 1

База: A[n+1...n] се състои от 0 най-големи елем. ✓

Поддръжка: Някое дост. на ред 1, което не е последно.

От инв. 2 имаме, че $j=n-i+1$, A[n-i+1] съдържа i-тият по гол.

От инв. 1 имаме, че A[n-i+2,...,n] съдържа i-1 тите по гол.

=> A[n-i+1....n] съдържа i-тите по гол.

$i' \leftarrow i+1$ $i = i'-1$ тогава при след. дост. на ред 1 A[n-i'+2 ...n] съдържа i-1-тите по гол. елем.

т.е. инв. се запазва

Терминация. Посл. достигане на ред 1 $i=n+1$

$A[n-n-1+2 .. n]$ ($A[1...n]$) съдържа $n+1-1$ (n)-тите нг елем. във възх. ред

Цикълът терминира при $i>n$ и няма други команди за изп => терминира

Произв. изп. на външния, разгл. вътр. цикъл

База: 1 дост на ред 2: $j=1$

$A[1...n+1-i]$ съдържа i-тият по гол. елем. на входа

От док. на инв. 1 $A[n+2-i...n]$ съд $i-1$ най-големи елем.

т.е i-тият по голем. се съдържа в $A[j...n+1-i]$ ($A[1...n+1-i]$)

Поддръжка: Доп., че инв 2 е изп. за някое дост на ред 2, което не е посл.

I сл: $A[j]>A[j+1]$ i-тият по гол е в $A[j...n+1-i]$ (от инв. 2)

при изп. на ред 4 се запазва $A[j+1...n+1-i]$ съдържа i-тия по гол.

Следващо изп. $j' \leftarrow j+1$ $j'=j-1$, тогава $A[j'-1+1 ... n+1-i]$ съд. i-тия по гол

II сл. $A[j] \le A[j+1]$ i-тият по гол. е в $A[j...n+1-i]$ (от инв 2)

$A[j]$ няма как да е i-тият по гол. т.к. подмасива $A[j+1...n+1-i]$ съд. по-големи

от него т.е. i-тият по гол. се съд. в $A[j+1, ..., n+1-i]$

Следващо изп. $j' \leftarrow j+1$ $j'=j-1$, тогава $A[j'-1+1 ... n+1-i]$ съд. i-тия по гол

Терм. $j=n-i+1$ т.е. $A[n-i+1 ... n-i+1]$ се съд. i-тият по гол.
$= A[n-i+1]$

заедно с док на инв. 1 $A[n-i+2...n]$ е сортиран във възх. ред т.е.

$A[n-i+1, n-i+2, ..., n]$ е сортиран във възх. ред

зад 2 GrahamScan($A[1...n]$ масив от числа) $P(s) := s.notEmpty$ && isEven / >10 / <0

```
1. s ← празен стек      O(1)
2. s.push(A[1])         O(1)
3. s.push(A[2])         O(1)
4.   for i←3 to n       O(n)
5.     while P(s)       O(n)
6.        s.pop()
7.     s.push(A[i])
```

$O(n) + O(n) + O(1) = O(n)$

pop push

зад3 RecAlg($a \in \mathbb{R}^+$, $n \in \mathbb{N}$) Връща a^n

```
1 if n=0 then
2     return 1
3 if n isEven then      (n≡0 mod 2)
4     return RecAlg(a*a, n/2)
5 return a* RecAlg(a, a-1)
```

база $n=0$ 1 ред е изп. и вр 1 ✓

$a^0 = 1$ ✓

ИП: $\forall m \le n$ е изп, че RecAlg$(a, m) = a^m$

ИС: Док $n+1 > 0$

I сл. $n+1$ е четно

1 ред е лъжа => 2 не се изп.

3 ред е истина => 4 ред се изп. RecAlg$(a*a, (n+1)/2)$

От ИП имаме $\forall m \le n$ RA връща a^m

$(n+1)/2 < n$ т.е. RecAlg$(a*a, (n+1)/2) = a^{2(\frac{n+1}{2})} = a^{n+1}$ ✓

II сл. 1 ред е лъжа => 2 не се изп.

3 ред е лъжа => 4 не се изп.

5 ред е истина => 6 ред се изп. $a*$ RecAlg$(a, (n+1)-1)$ $n \le n$

$a * \underbrace{a^n}_{ИП} = a^{n+1}$

зад 4. ALG(A[1...n]: масив от цели числа)

```
1. a ← 0
2. for i ← 1 to n
3.     for j ← 1 to n
4.         if i=j then
5.             for k ← 1 to n
6.                 a ← a+1
7. return a
```

$n.\sum_{k=1}^{n} 1 + (n^2-n).1 = n^2 + n^2 - n = O(n^2)$

зад5 ALG
1. for i←1 to n
2. for j←1 to n with step i
3. print ("a")

$$\sum_{i=1}^{n}\sum_{\substack{j=1\\ j+=i}}^{n} 1 \approx \sum_{i=1}^{n} \frac{n}{i} = n.\sum_{i=1}^{n} \frac{1}{n} = O(n\log n)$$

зад6 ALG
1. for i←1 to n
2. for j←1 to n with step i
3. for k←1 to n with step i
4. print("a")

$$\sum_{i=1}^{n}\sum_{\substack{j=1\\ j\pm i}}^{n}\sum_{\substack{k=1\\ k\pm i}}^{n} 1 = \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\pm i}}^{n}\left(\frac{n}{i}\right) =$$

$$= \sum_{i=1}^{n}\frac{n}{i}\sum_{\substack{j=1\\ j\pm i}}^{n} 1 = \sum_{i=1}^{n}\left(\frac{n}{i}\right)^2 = n^2\sum_{i=1}^{n}\frac{1}{i^2} =$$

$$= O(n^2)$$

зад7. ALG(A[1...n] : масив от цели числа)
1. curr ← A[1]
2. best ← A[1]
3. for i←2 to n
4. if A[i]+curr > A[i] then
5. curr ← curr + A[i]
6. else
7. curr ← A[i]
8. if curr > best
9. best ← curr
10. return best

1 -1 2 -2
curr
best

KADANE: връща най-голяма сума на непр. подмасив на A[1...n]

Инвариант: при всяко достигане на ред 3 best съдържа най-голяма сума на подмасив A[1...i-1], curr съдържа най-голяма сума на A[1...i-1] завършваща в i-1

зад. 53 сини 42 червени
while не остане 1 топка, извади 2 случайни
if цвят 1 ≠ цвят 2 добави 1 червена
else добави 1 синя
return оставаща топка