Асимптотика та складність

Задача 1
Знайти суму елементів масиву

```
for(int i = 0; i < n; ++i)
    R = R + a[i];
```

Додавання	n
Присвоєння	n
Загалом	2n

Задача 2
Знайти максимум масиву

```
for(int i = 0; i < n; ++i)
    if(a[i] > R)
        R = a[i];
```

Порівняння	n
Присвоєння	від 1 до n
Загалом	Від n + 1 до 2n

Задача 3

Знайти

$$\sum_{i=0}^{n-1} \sum_{j=0, i!=j}^{n-1} a[i]a[j]$$

Розв’язок 1

```
for(int i = 0; i < n; ++i)
    for(int j = 0; j < n; ++j)
        if(i != j)
            R = R + a[i]*a[j];
```

Операцій загалом

$$n\ (\ 3(n\ -1) + n\ ) =$$

$$= n\ (\ 4n - 3\ )$$

	при i = 0	при i = 1	...
Порівняння	n	n	...
Додавання, множення і присвоєння	n - 1	n - 1	...
Загалом	3(n - 1) + n	3(n - 1) + n	...

Розв’язок 2

a[i]	**1**	**5**	**2**	**-4**	**1**	**6**
1	*1*	5	2	-4	1	6
5	5	*25*	10	-20	5	30
2	2	10	*4*	-8	2	12
-4	-4	-20	-8	*16*	-4	-24
1	1	5	2	-4	*1*	6
6	6	30	12	-24	6	*36*

Розв'язок 2

```
for(int i = 0; i < n; ++i)
    for(int j = i + 1; j < n; ++j)
        R = R + a[i]*a[j];
R = R * 2;
```

Операцій загалом

$3(n - 1) + 3(n - 2) + ... + 3 + 1 =$
$= (3/2) * n(n - 1) + 2$

	при i = 0	при i = 1	...
Додавання, множення і присвоєння	n -1	n - 2	...
Загалом	3(n - 1)	3(n - 2)	...

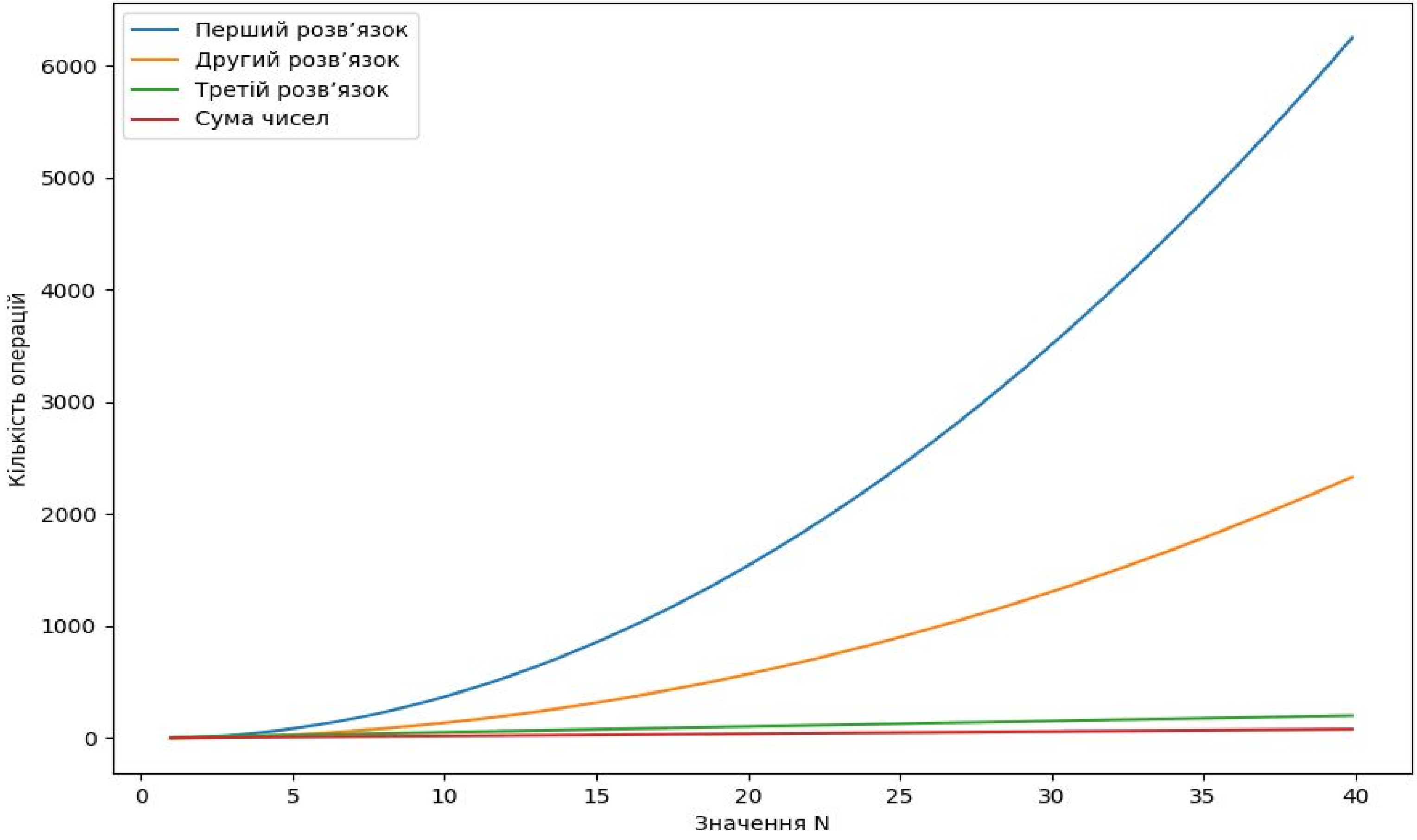
Розв’язок 3

$$\sum_{i=0}^{n-1}\sum_{j=0,i!=j}^{n-1} a[i]a[j] = \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} a[i]a[j] - \sum_{i=0}^{n-1} a[i]^2 = \sum_{i=0}^{n-1} a[i]\sum_{j=0}^{n-1} a[j] - \sum_{i=0}^{n-1} a[i]^2 = \sum_{i=0}^{n-1} a[i]S - \sum_{i=0}^{n-1} a[i]^2 = S^2 - \sum_{i=0}^{n-1} a[i]^2$$

```
for(int i = 0; i < n; ++i){
    sum = sum + a[i];
    sqSum = sqSum + a[i]*a[i];
}
result = sum*sum - sqSum;
```

Додавання/віднімання	2n + 1
Множення	n + 1
Присвоєння	2n + 1
Загалом	5n + 3

	N = 5	N = 10	N = 100	N = 1000	N =10000	N = 100000	N = 1000000
Рішення 1 n*(4n - 3)	85	370	39700	3’997’000	399’970’000	39’999’700’000	3’999,997’000’000
Рішення 2 3/2*n*(n-1)+2	32	137	14852	1’498’502	149’985’002	14,999’850’002	1’499,998’500’002
Рішення 3 5n + 3	28	53	503	5’003	50’003	500’003	5’000’003
Пошук суми 2n	10	20	200	2’000	20’000	200’000	2’000’000


Перший розв’язок
Другий розв’язок
Третій розв’язок
Сума чисел
Кількість операцій
Значення N
0
1000
2000
3000
4000
5000
6000
0
5
10
15
20
25
30
35
40

Означення

$T(n) = O(f(n))$ - існує деяке додатнє число **c**, таке що для всіх n починаючи з деякого виконується $T(n) \le cf(n)$

$$n * (4n - 3) \le 4 * n^2 \implies n * (4n - 3) = O(n^2)$$

$$\frac{3}{2}n(n-1) + 1 \le \frac{3}{2}n^2 \implies \frac{3}{2}n(n-1) + 1 = O(n^2)$$

$$5 * n + 3 \le 6 * n (n \ge 3) \implies 5 * n + 3 = O(n)$$

$$2n \le 2n \implies 2n = O(n)$$

Вправа 1

```
int j = 0;
for(int i = 0; i < n; ++i)
    while(j < n) {
        j = j + 1;
        R = R + 1;
}
```

Додавання	2*n
Присвоєння	2*n
Порівняння	(n+1)+ (n-1) = 2*n
Загалом	6n

Складність: $T(n) = O(n)$

Вправа 2

```
while(n > 1) {
    n = n / 2;
    r = r + 1;
}
```

Складність:

$T(n) = O(\log n)$

Властивості

$$O(cf(n)) = O(f(n))$$

$$O(3n^2) = O(n^2)$$

$$f(n) = O(f(n))$$

$$n = O(n), n^2 = O(n^2),$$

$$O(f(n) + g(n)) = O(f(n)), f(n) > g(n), n > n_0$$

$$O(n + n^2) = O(n^2)$$

$$f(n) * O(g(n)) = O(f(n) * g(n))$$

$$n * O(n^2) = O(n^3)$$

$$O(f(n)) + O(g(n)) = O(f(n) + g(n))$$

$$O(n) + O(n) = O(2n) = O(n)$$

Асимптотики для більш ніж однієї змінної

(на прикладі двох змінних)

```
for(int i=0;i<n;++i) {
    for(int j=0;j<m;++j) {
        r = r + a[i]*b[j];
    }
}
```

Складність:

O(m*n)

Асимптотики для більш ніж однієї змінної

(приклад 2)

```
for(int i = 0; i < n;++i){
    /*
    Деякий код з асимптотикою O(m^2)
    */
}
```

Складність: O(n * m^2)

Асимптотики для більш ніж однієї змінної

(приклад 3)

```
for(int i = 0; i < n; ++i){
    /*
    Деякий код з асимптотикою O(m + n)
    */
}
```

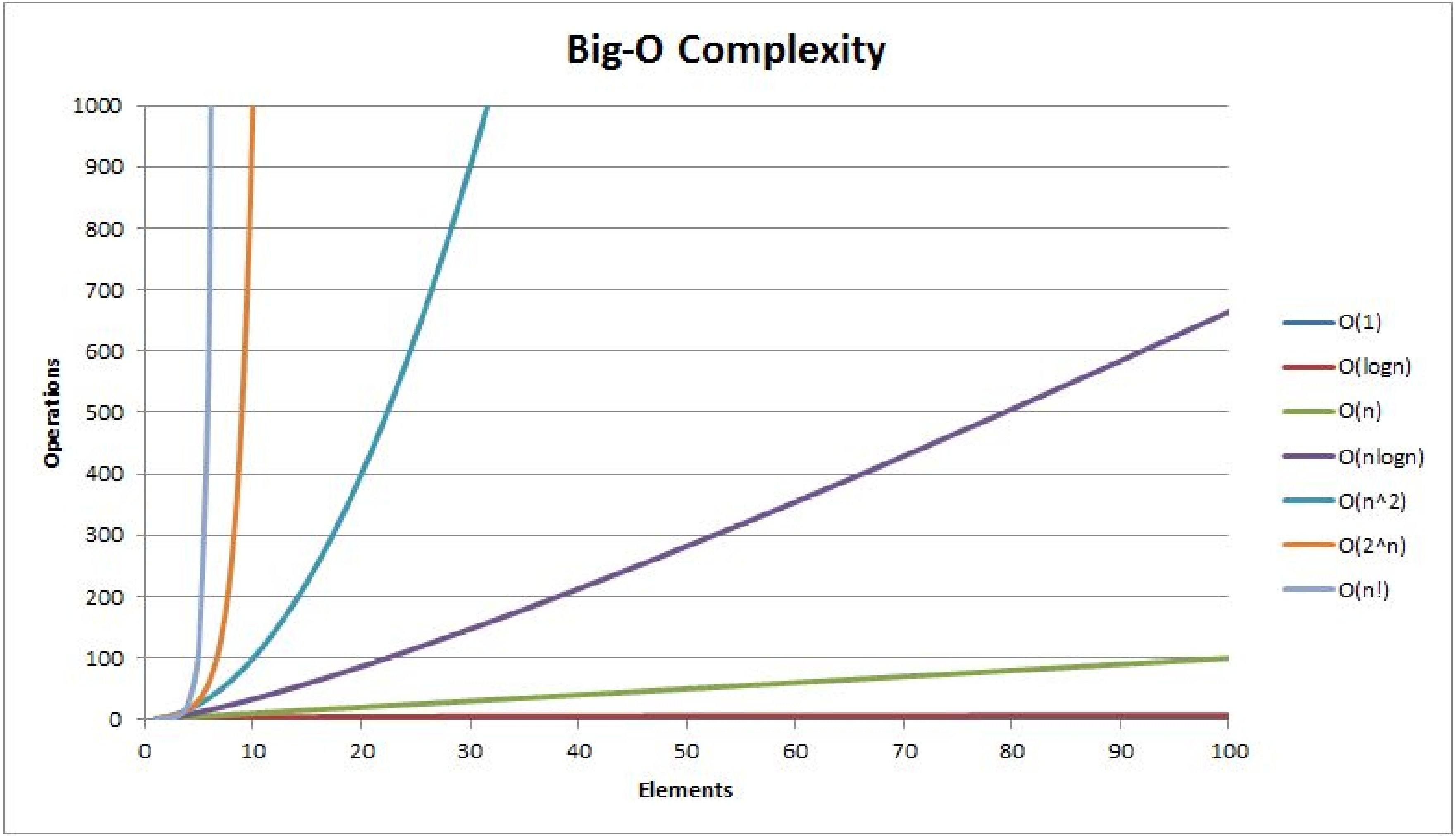
Складність: O(n * (m + n))

Асимптотики для більш ніж однієї змінної

(приклад 4)

```
for(int i = 0; i < n; ++i){
    /*
    Деякий код з асимптотикою O(m + i)
    */
}
```

$$\sum_{i=0}^{n-1} O(m+i) = O(mn + \sum_{i=0}^{n-1} i) = O(mn + \frac{n(n-1)}{2}) = O(mn + n^2)$$


Big-O Complexity
1000
900
800
700
600
500
400
300
200
100
0
Operations
0
10
20
30
40
50
60
70
80
90
100
Elements
O(1)
O(logn)
O(n)
O(nlogn)
O(n^2)
O(2^n)
O(n!)