2.11.
```
def g(n)
    sum = 0                          O(1)
    for i in range (1, n+1):         O(n)
        sum = sum + i + f(i)         $\sum_{i=1}^{n} O(i) \leq n \cdot O(n) = O(n^2)$
    return sum                       O(1)
```

- Асимптотична оцінка виконання функції $g(n)$ у найгіршому

випадку: $T_g(n) = O(n^2)$.

- Результат функції: $\sum_{i=1}^{n} i + \sum_{i=1}^{n} f(i) = \dfrac{n(n+1)}{2} + \sum_{i=1}^{n} \left(\sum_{k=1}^{i} k \right) =$

$= \dfrac{n(n+1)}{2} + \sum_{i=1}^{n} \dfrac{i(i+1)}{2} = \dfrac{n(n+1)}{2} + \dfrac{1}{2} \left(\sum_{i=1}^{n} i + \sum_{i=1}^{n} i^2 \right) =$

$= \dfrac{3 \cdot n(n+1)}{4} + \dfrac{n(n+1)(2n+1)}{2 \cdot 6} = \dfrac{3n(n+1)}{4} + \dfrac{n(n+1)(2n+1)}{12} = \dfrac{n(n+1)(2n+10)}{12} =$

$= \dfrac{n(n+1)(n+5)}{6}$

- Функцію можна оптимізувати:

```
def g(n):
    ~return~ 3n(n+1)/4 + n(n+1)(2n+1)/12
    return n·(n+1)(n+5)/6
```
Тоді $T_g(n) = O(1)$.