Задача 0.1. *Да разгледаме следната функция:*

```
Bubble(A[1..n]: array of integers; m: index in A)
@1    j ← 1
@2    for i = 1 to m − 1 do
@3       if A[i] > A[i + 1] then
@4          swap(A, i, i + 1)
@5          j ← i
@6    done
@7    return j
```

Да се докаже, че:

1. *ако* j *е резултатът от изпълнението на функцията* $Bubble(A, m)$*, а* $A'[1..n]$ *е модифицираният масив* A*, то:*
   - $A'[j + 2..n] = A[j + 2.n]$ *и* $A'[1..j + 1]$ *е пермутация на масива* $A[1..j + 1]$.
   - $A'[j..m]$ *е сортиран във възходящ ред.*
   - $A'[j + 1] > A[i]$ *за всяко* $i \leq j$.
2. *времевата сложност на алгоритъма* $Bubble(A, m)$ *е* $\Theta(m)$.
3. *ако при изпълнението на* $Bubble(A, m)$*, масивът* A *се е изменил до* A'*, а ред @4 се е изпълнил* t *пъти, то:*

   $$|\sigma(A, m)| = |\sigma(A', m)| + t,$$

   където $\sigma(A, m)$ *и* $\sigma(A', m)$ *са съответно инверсиите в* $A[1..m]$ *и* $A'[1..m]$.

Да разгледаме сега следната функция, която използва $Bubble$.

```
BubbleSort(A[1..n]: array of integers)
@1    m ← n
@2    while m > 1 do
@3       m ← Bubble(A, m)
@4    done
```

Да се докаже, че $BubbleSort(A)$ *модифицира масива* $A[1..n]$ *до* $A'[1..n]$*, който е сортирана пермутация на* A*. Нещо повече времевата сложност на* $BubbleSort$ *е* $O(n^2)$.

Задача 0.2. *Нека* $MinMax(A[1..n])$ *е детерминиран алгоритъм, който по даден масив от числа* $A[1..n]$ *връща наредена двойка* (i, j)*, за която* $i = \arg\min A[k]$ *и* $j = \arg\max A[k]$*. Ако в* $MinMax$ *елементите на масива* A *участват единствено в сравнения по между си, да се докаже, че в лошия случай* $MinMax$ *прави поне* $\frac{3n}{2} - 2$ *такива сравнения. Да се покаже алгоритъм* $MinMax$*, който постига тази сложност при всеки вход от вида* $A[1..2^k]$.

Задача 0.3. *Нека* $f, g, h : \mathbb{N} \to \mathbb{R}^+$ *са тотални функции.*

1. *Нека* $F : \mathbb{N} \to \mathbb{R}^+$ *е функцията:*

   $$F(n) = \max\{f(m) \mid m \leq n\}.$$

   Да се докаже, че F *е монотонно растяща и* $f \in O(F)$*. Винаги ли е вярно, че* $F \in O(f)$*?*
2. *Ако* $f, h \in O(g)$*, да се докаже, че* $f + h \in O(g)$ *и* $fh \in O(g^2)$.
3. *Нека* $F : \mathbb{N} \to \mathbb{R}^+$ *и* $G : \mathbb{N} \to \mathbb{R}^+$ *са дефинирани като:*

   $$F(n) = \sum_{k=1}^{[h(n)]} f(k) \text{ и } G(n) = \sum_{k=1}^{[h(n)]} g(k).$$

   Да се докаже, че ако $f \in O(g)$*, то* $F \in O(G + h)$.
4. *Често се използва означението* $2^{O(f)} = \bigcup\{O(2^g) \mid g \in O(f)\}$*. Вярно ли е, че* $O(2^f) \subseteq 2^{O(f)}$*? Вярно ли е, че* $2^{O(f)} \subseteq O(2^f)$.

Упътване 0.1. 1. Забележете, че m не се променя в рамките на $Bubble$.

- Нека A_i и j_i са състоянията на масива A и променливата j непосредствено преди изпълнението на i-тата итерация на for-цикъла от ред @2. Формулирайте подходящ инвариант $I(i)$, който да отразява конюнкцията на свойствата, които искаме да докажем, но *релативизирайте* m. Тоест, в момента i няма как да е изпълнено второто свойство в общия случай, защото не сме разгледали елементите от $A[i+1..m]$, но за елементите $A[j_i..i]$, които сме разгледали може да се надяваме, че са сортирани във възходящ ред. Докажете, че $I(i)$ е вярно за всяко $i \leq m-1$ с индукция по i.
- Аргументирайте, че в тялото на for-цикъла се изпълняват константен брой атомарни операции и общият брой итерации е m.
- Аргументирайте, че при всяко изпълнение на ред @4 броят на инверсиите се намалява с 1 и че той не се променя никъде освен на ред @4.

2. Докажете, че ако $m' = Bubble(A, m)$ и $n \geq m > 1$, то $1 \leq m' < m - 1$. Обосновете, че $BubbleSort(A)$ завършва за не повече от n итерации на $while$-цикъла.

3. Ако A_i и m_i са състоянието на входния масив $A[1..n]$ и променливата m преди i-тата итерация на while-цикъла, формулирайте инвариант:

$$I(i) : A_i[1..n] \text{ е пермутация на } A[1..n] \text{ и } A_i[m_i..n] \text{ е сортиран във възходящ ред .}$$

Докажете, че $I(i)$ е истина за всяко i и оттук заключете, че BubbleSort коректно сортира масива A.

4. Използвайте първата част на задачата, за да аргументирате, че всяка итерация на while-цикъла се изпълнява за време $\Theta(m)$ и завършете като обосновете, че различните итерации на цикъла съответстват на различни стойности на m в множеството $\{1, 2, \ldots, n\}$.

Упътване 0.2. 1. В зависимост от заявките за сравнения, които прави $MinMax$ върху масива $A[1..n]$ ще построим ориентиран ацикличен граф $G = (V, E)$, където $V = \{1, 2, \ldots, n\}$, а

$$E = \{(i, j) \mid \text{ направена е заявка за } A[i] \text{ и } A[j] \text{ и отговорът е } A[i] < A[j]\}.$$

2. Покажете, че ако има такъв граф, то всяко топологическо сортиране на G е съгласувано със заявките, които е направил алгоритъмът.

3. Заключете, че в такъв граф G има единствени върхове u, v, за които $d^-(u) = d^+(v) = 0$.

4. Графа G строим на стъпки в зависимост от заявките на $MinMax$. В началото $G_k = (V, E_k = \emptyset)$. Това, което ни интересува са множествата:

$$U_k^+ = \{u \in V \mid d_{G_k}^+(u) = 0\} \text{ и } U_k^- = \{u \in V \mid d_{G_k}^-(v) = 0\}.$$

Целта е $|U_k^+| + |U_k^-|$ да намалява бавно. Тъй като $|U_0^+| + |U_0^-| = 2n$, а в края $|U_t^+| + |U_t^-| = 2$, то t, а оттам и броят на ребрата на G ще трябва да бъде голям.

5. За реализацията тази идея, разгледайти няколко случая за заявка (i, j) и съответно ребро, което трябва да се добави към E_k, за да се получи E_{k+1}:

   (а) $i \to_{G_k}^* j$. Съобразете, че тогава единствено реброто (i, j) може да се добавим към E_{k+1}, но това не променя $U_{k+1}^- = U_k^-$ и $U_{k+1}^+ = U_k^+$.

   (б) ако $i, j \in U_k^+ \cap U_k^-$, то каквото и ребро да добавим U_k^- и U_k^+ намаляват с по един елемент 1. Тази случай обаче може да настъпи не повече от $\frac{n}{2}$ пъти! (Защо?)

   (в) Докажете, че във всички останали случаи може да добавите реброто (i, j) или (j, i) към E_k така, че $|U_k^-| + |U_k^+| \leq |U_{k+1}^-| + |U_{k+1}^+| + 1$.

6. Пребройте колко стъпки от последния тип са необходими и довършете.

Упътване 0.3.

Заместете в дефиницията и забележете, че $f(n) \leq F(n)$ за всяко n. За втория въпрос, разгледайте функция f, която расте неограничено върху подмножество $A \subsetneq \mathbb{N}$ и е нула (или друга константа) върху $\mathbb{N} \setminus A$.

Приложете дефиницията и ако $f(n) \leq c_1 g(n)$ за всяко $n > n_1$, а $h(n) \leq c_2 g(n)$ за всяко $n > n_2$, забележете, че за всяко $n > \max(n_1, n_2)$ е в сила неравенството:

$$f(n) + h(n) \leq (c_1 + c_2)g(n).$$

Завършете, като обясните, как оттук следва, че е изпълнена дефиницията за $f + h \in O(g)$.

1. Разгледайте n_1 и $c > 0$, за които $f(n) \leq cg(n)$ за $n > n_1$.
2. Обозначете с $c_0 = \max\{f(m) \,|\, m \leq n_1\}$.
3. Разделете сумата $\sum_{k=1}^{[h(n)]} f(k)$ на две части:
$$S_1 = \sum_{k=1}^{\min(n_1,[h(n)]} f(k) \text{ и } S_2 = \sum_{k=\min(n_1,[h(n)])+1}^{[h(n)]} f(k).$$
4. Докажете, че $S_1 \leq c_0 h(n)$.
5. Докажете, че $S_2 \leq c \sum_{k=\min(n_1,[h(n)])+1}^{[h(n)]} g(k)$.
6. Довършете, като покажете как от горните две следва, че дефиницията $F \in O(G + h)$ е изпълнена.

Тъй като $f \in O(f)$, то първото твърдение следва. За втората част забележете, че $3^n \in 2^{O(n)}$.