Занятие 2

Доказательство корректности алгоритма, использованного нами на предыдущем занятии в задаче D. Алена и mex

Еще раз напомню решение:

Сортируем по возрастанию a_i. Давай будем держать в уме mex, который мы не можем набрать (изначально $mex = 1$, минимальное возможное значение).

Будем проходиться по массиву a. Мы хотим сделать $b_i = mex$, чтобы увеличить минимальный mex, который мы не можем набрать.

Когда мы можем присвоить $b_i = mex$? Поскольку мы знаем, что $b_i \leq a_i$, мы сможем присвоить $b_i = mex$ тогда и только тогда, когда $a_i \geq mex$. Если $a_i \geq mex$, тогда увеличим $mex = mex + 1$ и пойдем дальше по массиву (к a_{i+1}).

Давай посмотрим, почему нам выгодно выполнять этот алгоритм в порядке возрастания a.

Посмотрим на какие-то два индекса i, j $(i < j)$. Давай докажем от противного, предположим, что $a_i > a_j$ (напомню, eventually мы хотим показать, что $a_i \leq a_j$ оптимальнее). Пусть наш массив состоит только из чисел a_i, a_j. На сколько может максимально увеличиться mex? Давай разберем 3 случая:

1. $mex > a_i > a_j$. Тогда мы не сможем увеличить mex. Если мы поменяем местами a_i, a_j хуже не станет - мы все так же не сможем увеличить mex.
2. $a_i \geq mex \geq a_j$. Тогда мы сможем увеличить mex на единицу после a_i. Однако потом увеличить mex мы не сможем, поскольку $mex \geq a_j \Leftrightarrow mex + 1 > a_j$. В свою очередь если бы мы поменяли местами a_i, a_j, после обработки a_j мы бы увеличили mex на единицу и, возможно, смогли бы увеличить mex на единицу после a_i.
3. Аналогично рассматриваем оставшийся случай.

В конце концов понимаем, что если $a_i > a_j$ при $i < j$, то нам КАК МИНИМУМ НЕ ХУЖЕ поменять местами a_i, a_j, поэтому просто сделаем это и запустим алгоритм выше.

Это не очень формальное доказательство, впоследствии мы научимся более строго доказывать жадные алгоритмы.

Код:

```cpp
#include <bits/stdc++.h>

int main() {
    int n;
    std::cin >> n;

    std::vector<int> a(n);
```

```
    for (int i = 0; i < n; ++i) {
        std::cin >> a[i];
    }

    std::sort(a.begin(), a.end());
    int mex = 1;
    for (int el : a) {
        if (el >= mex) {
            ++mex;
        }
    }

    std::cout << mex << '\n';
}
```

> Ссылка на условия: https://algocourses.ru/files/course_bp2022/contest-29851-ru.pdf

> Ссылка на контест на этой странице: https://algocourses.ru/bp2022/

1. I
2. J
3. K

> Ссылка на контест: https://codeforces.com/group/jtU6D2hVEi/contest/105000

1. D

Считывание массивов в строку (осторожно, Python!):

```
with open("input.txt") as file_in:
    a, b = [], []
    for line in file_in:
        if len(a) == 0:
            a = [int(x) for x in line.rstrip('\n').split()]
        else:
            b = [int(x) for x in line.rstrip('\n').split()]

def merge(a, b):
    arr = []
    i, j = 0, 0
    while i < len(a) or j < len(b):
        if j >= len(b) or (i < len(a) and a[i] < b[j]):
            arr.append(a[i])
            i += 1
        else:
            arr.append(b[j])
            j += 1
```

```
    return arr

with open("output.txt", "w") as file_out:
    print(*merge(a, b), file=file_out)

```

2. L

Код (reference only!):

```cpp
#include <bits/stdc++.h>

bool comparator(int a, int b) {
    return (a % 10) < (b % 10);
}

int main() {
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    int n;
    std::cin >> n;

    std::vector<int> a(n);
    for (int i = 0; i < n; ++i) {
        std::cin >> a[i];
    }

    std::stable_sort(a.begin(), a.end(), comparator);

    for (int x : a) {
        std::cout << x << ' ';
    }
    std::cout << '\n';
}
```