Ordenação
Professor Wladimir

29 de abril de 2025

75. Sort Colors

url: https://leetcode.com/problems/sort-colors/description/

Given an array nums with n objects colored red, white, or blue, sort them in-place so that objects of the same color are adjacent, with the colors in the order red, white, and blue.

We will use the integers 0, 1, and 2 to represent the color red, white, and blue, respectively.

You must solve this problem without using the library's sort function.

Example 1:

Input: nums = [2,0,2,1,1,0]

Output: [0,0,1,1,2,2]

Example 2:

Input: nums = [2,0,1]

Output: [0,1,2]

```cpp
void sortColors(vector<int>& nums) {
    int freq[3] = {0,0,0};

    for(int x : nums){
        freq[x]++;
    }
    int pos = 0;
    for(int i = 0; i < 3; i++)
        for(int j = 0; j < freq[i]; j++){
            nums[pos] = i;
            pos++;
        }

}
```

```cpp
void sortColors(vector<int>& nums) {
        int pos = 0;
        int n = nums.size();
        //Colocando zeros no inicio do vetors nums[0..n-1]
        for(int i = 0; i < n; i++){
            if(nums[i] == 0) swap( nums[pos++], nums[i]);
        }
        //colocanco uns no inicio do vetor nums[pos..n-1]
        for(int i = pos; i < n; i++){
            if(nums[i] == 1) swap( nums[pos++], nums[i]);
        }
}
```

Referência:

1. Ordenação Linear: Counting sort https://joaoarthurbm.github.io/eda/posts/ordenacao-linear/

2037. Minimum Number of Moves to Seat Everyone

url: https://leetcode.com/problems/minimum-number-of-moves-to-seat-everyone/description/

There are n availabe seats and n students standing in a room. You are given an array seats of length n, where seats[i] is the position of the ith seat. You are also given the array students of length n, where students[j] is the position of the jth student.

You may perform the following move any number of times:

Increase or decrease the position of the ith student by 1 (i.e., moving the ith student from position x to x + 1 or x - 1) Return the minimum number of moves required to move each student to a seat such that no two students are in the same seat.

Note that there may be multiple seats or students in the same position at the beginning.

```cpp
int minMovesToSeat(vector<int>& seats, vector<int>& students) {
    sort( seats.begin(), seats.end() );
    sort( students.begin(), students.end() );
    int moves = 0;
    int n = seats.size();
    for( int i = 0; i < n; i++){
        moves += abs(seats[i]-students[i]);
    }
    return moves;
}
```

Leituras recomendadas:

1. Método Guloso https://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/guloso.html
2. Algoritmos gulosos: definições e aplicações https://www.ic.unicamp.br/~rocha/msc/complex/algoritmosGulososFinal.pdf
3. Algoritmos Gulosos https://www.lia.ufc.br/~wladimir/gemp/aulas/Algoritmos%20Gulosos%20-%20OBI%202003.pdf
4. Material didático sobre algoritmos gulosos https://linux.ime.usp.br/~colombo/mac0499/monografia.pdf

Vídeo Algoritmo gulosos https://www.youtube.com/watch?v=-zlOUxwEaMc&list=PLPpVvPOjdervYWBxB9wrXy9index=4

5. Canonical Coin Systems for Change-Making Problems https://arxiv.org/pdf/0809.0400
6. What This Country Needs is an 18¢ Piece https://cs.uwaterloo.ca/~shallit/Papers/change2.pdf

2094. Finding 3-Digit Even Numbers

URL: https://leetcode.com/problems/finding-3-digit-even-numbers/description/

You are given an integer array digits, where each element is a digit. The array may contain duplicates.

You need to find all the unique integers that follow the given requirements:

The integer consists of the concatenation of three elements from digits in any arbitrary order. The integer does not have leading zeros. The integer is even. For example, if the given digits were [1, 2, 3], integers 132 and 312 follow the requirements.

Return a sorted array of the unique integers.

Example 1:

Input: digits = [2,1,3,0]

Output: [102,120,130,132,210,230,302,310,312,320]

Explanation: All the possible integers that follow the requirements are in the output array.

Notice that there are no odd integers or integers with leading zeros.

Example 2:

Input: digits = [2,2,8,8,2]

Output: [222,228,282,288,822,828,882]

Explanation: The same digit can be used as many times as it appears in digits.

In this example, the digit 8 is used twice each time in 288, 828, and 882.

```cpp
vector<int> findEvenNumbers(vector<int>& digits) {

    set <int> numeros;

    for(int i = 0; i < digits.size(); i++){
        for(int j = 0; j < digits.size(); j++){
            for(int k = 0; k < digits.size(); k++){
                if(i == j || j == k || i == k) continue;
                int n = digits[i]*100 + digits[j]*10 + digits[k];
                if(n%2==0 && n >= 100)
                    numeros.insert(n);
            }
        }
    }
    vector <int> res;
    for(int x : numeros) res.push_back(x);
    return res;
}
```

Exercícios

1. 1984. Minimum Difference Between Highest and Lowest of K Scores https://leetcode.com/problems/minimum-difference-between-highest-and-lowest-of-k-scores/description/