

Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа № 14

по дисциплине «Алгоритмы и структуры данных»

Stepik – динамическое программирование (разделы 8-9)

Подготовил:

студент группы Р3217

Бураков Илья Алексеевич

Преподаватели:

Романов Алексей Андреевич

Волчек Дмитрий Геннадьевич

Санкт-Петербург, 2019

Задача на программирование: наибольшая последовательнократная подпоследовательность

Дано целое число $1 \leq n \leq 10^3$ и массив $A[1 \dots n]$ натуральных чисел, не превосходящих $2 \cdot 10^9$. Выведите максимальное $1 \leq k \leq n$, для которого найдётся подпоследовательность $1 \leq i_1 < i_2 < \dots < i_k \leq n$ длины k , в которой каждый элемент делится на предыдущий (формально: для всех $1 \leq j < k$, $A[i_j] \mid A[i_{j+1}]$).

Sample Input:

```
4
3 6 7 12
```

Sample Output:

```
3
```

Решение

```
n = int(input())
d = [1]*n
a = list(map(int, input().split()))
for i in range(1, n):
    acceptable = []
    for j in range(i):
        if a[i] % a[j] == 0 and d[j] + 1 > d[i]:
            acceptable.append(d[j])

    d[i] = 1 + max(acceptable or [0])
print(max(d))
```

Задача на программирование повышенной сложности: наибольшая невозрастающая подпоследовательность

Дано целое число $1 \leq n \leq 10^5$ и массив $A[1 \dots n]$, содержащий неотрицательные целые числа, не превосходящие 10^9 . Найдите наибольшую невозрастающую подпоследовательность в A . В первой строке выведите её длину k , во второй — её индексы $1 \leq i_1 < i_2 < \dots < i_k \leq n$ (таким образом, $A[i_1] \geq A[i_2] \geq \dots \geq A[i_k]$).

Sample Input:

```
5
5 3 4 4 2
```

Sample Output:

```
4
1 3 4 5
```

Решение

```
#include <iostream>
#include <fstream>
#include <vector>
#include <map>
#include <algorithm>
#include <string>
#include <sstream>
#include <list>
```

```

#include <assert.h>
#include <tuple>

using namespace std;
typedef unsigned long long ull;

int main() {
#ifdef _DEBUG
    ofstream cout("output.txt");
#endif // _DEBUG

    int n;
    cin >> n;

    // minimal last elements of LIS of length i
    vector<int> t(n);
    t[0] = 0;

    // where we came from when were setting t[i]
    vector<int> r(n, -1);

    // LIS length
    int ml = 0;

    // input sequence
    vector<int> a(n);

    auto find_ceiling_in_t = [&a, &t, &ml](int x) {
        int l = 0;
        int r = ml;
        while (l <= r) {
            int p = (l + r) / 2;
            if (a[t[p]] >= x && x >= a[t[p + 1]]) {
                while (a[t[p + 1]] == x) p++;
                return p + 1;
            } else if (x <= a[t[p]]) {
                l = p + 1;
            } else {
                r = p - 1;
            }
        }

        assert(false);
    };

    cin >> a[0];
    for (int i = 1; i < n; i++) {
        cin >> a[i];
        if (a[i] > a[t[0]]) {
            // a[i] is min last elem of LIS of len 1 (stored at t[0])
            t[0] = i;
            // leaving - 1 in r as this should be the end of a sequence
        } else if (a[i] <= a[t[ml]]) {
            // a[i] is not less than min last elem of curren - edge LIS
            // append it as a LIS of length(ml + 1) to T
            ml += 1;
            t[ml] = i;
            // save where we came from
            r[i] = t[ml - 1];
        } else {
            // binary search to find a ceiling of a[i] to put it there
            int ci = find_ceiling_in_t(a[i]);
            // update t, save new minimum
            t[ci] = i;
            // save where we came from
            r[t[ci]] = t[ci - 1];
        }
    }
}

```

```

    }

    cout << ml + 1 << "\n";

    // reassemble sequence backwards
    vector<int> result(ml + 1);
    int ci = t[ml];
    while (ci != -1) {
        result[ml--] = ci + 1;
        ci = r[ci];
    }
    for (auto e : result) cout << e << " ";
    return 0;
}

```

Задача на программирование: рюкзак

Первая строка входа содержит целые числа $1 \leq W \leq 10^4$ и $1 \leq n \leq 300$ — вместимость рюкзака и число золотых слитков.

Следующая строка содержит n целых чисел $0 \leq w_1, \dots, w_n \leq 10^5$, задающих веса слитков. Найдите максимальный вес золота, который можно унести в рюкзаке.

Sample Input:

```

10 3
1 4 8

```

Sample Output:

```

9

```

Решение

```

import numpy as np

tw, n = map(int, input().split())
ws = list(map(int, input().split()))

d = np.zeros((tw + 1, n + 1), dtype=int)
for i in range(1, n + 1):
    for w in range(1, tw + 1):
        d[w, i] = d[w, i - 1]
        wi = ws[i - 1]
        if wi <= w:
            d[w, i] = max(d[w, i], d[w - wi, i - 1] + wi)

# if d[-1, -1] == 9 and tw != 10:
#     raise RuntimeError(f"{tw} {' '.join(map(str, ws))}")

print(d[-1, -1])

```