

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЕТ
ПО ПРАКТИЧЕСКОЙ РАБОТЕ №11
дисциплины «Алгоритмизация»

Выполнил:

Лейс Алексей Вячеславович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем»

(подпись)

Руководитель практики кандидат тех.
наук доцент кафедры
инфокоммуникаций: Воронкин Р.А

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2023 г.

Порядок выполнения работы:

Динамическое программирование

Решение Фибоначчи из примера в видео

```
1 using System;
2
3 class Program
4 {
5     static void Main()
6     {
7         int N = 10;
8         Console.WriteLine($"Фибоначчи({N}) разными функциями:");
9         Console.WriteLine($"fib_td({N}) = {Fibonacci(N, "TD")}");
10        Console.WriteLine($"fib_bu({N}) = {Fibonacci(N, "BU")}");
11        Console.WriteLine($"fib_bu_improved({N}) = {Fibonacci(N, "BU_I")}");
12    }
13
14    static int Fibonacci(int n, string func = "TD")
15    {
16        int[] f = new int[n + 1];
17
18        int FibTD(int k)
19        {
20            if (k <= 1)
21                f[k] = k;
22            else
23                f[k] = FibTD(k - 1) + FibTD(k - 2);
24            return f[k];
25        }
26
27        int FibBU(int k)
28        {
29            int[] fib = new int[k + 1];
30            fib[0] = 0;
31            fib[1] = 1;
32            for (int i = 2; i <= k; i++)
33                fib[i] = fib[i - 1] + fib[i - 2];
34            return fib[k];
35        }
36
37        int FibBUImproved(int k)
38        {
39            if (k <= 1)
40                return k;
41            int prev = 0, curr = 1;
42            for (int i = 1; i < k; i++)
43            {
44                int temp = curr;
45                curr = prev + curr;
46                prev = temp;
47            }
48            return curr;
49        }
50
51        switch (func)
52        {
53            case "TD":
54                f = new int[n + 1];
55                return FibTD(n);
56            case "BU":
57                return FibBU(n);
58            case "BU_I":
59                return FibBUImproved(n);
60            default:
61                Console.WriteLine($"Неизвестная функция {func}");
62                Environment.Exit(1);
63                return 0; // Эта строка не выполняется, но нужна для компиляции
64        }
65    }
66 }
67
```

Выбрать Консоль отладки Microsoft Visual Studio

Фибоначчи(10) разными функциями:
fib_td(10) = 55
fib_bu(10) = 55
fib_bu_improved(10) = 55

I:\ИВТ-6-о-22-1\Алгоритмизация\11\ConsoleApp1\

Чтобы автоматически закрывать консоль при остано
томатически закрыть консоль при остановке отла
Нажмите любую клавишу, чтобы закрыть это окно:

Решения задачи о рюкзаке в двух случаях: когда предметов неограниченное количество, и когда каждый предмет может быть использован только один раз

```
class Program
{
    static void Main()
    {
        int W = 10;
        int[] weight = { 6, 3, 4, 2 };
        int[] cell = { 30, 14, 16, 9 };

        var result = KnapsackBU(W, weight, cell);
        Console.WriteLine($"С повторениями: {result.WithRep} \n Без повторений: {result.WithoutRep.Value}");
        Console.WriteLine("Реконструированное решение: [" + string.Join(", ", result.WithoutRep.Solution) + "]");
    }

    static (int WithRep, (int Value, int[] Solution) WithoutRep) KnapsackBU(int W, int[] weight, int[] cell)
    {
        int KnapsackWithReps(int w, int[] weight, int[] cell)
        {
            int[] d = new int[W + 1];
            for (int i = 1; i <= W; i++)
            {
                for (int j = 0; j < weight.Length; j++)
                {
                    if (weight[j] <= i)
                        d[i] = Math.Max(d[i], d[i - weight[j]] + cell[j]);
                }
            }
            return d[W];
        }

        (int, int[]) KnapsackWithoutReps(int W, int[] weight, int[] cell)
        {
            int[] d = new int[W + 1, weight.Length + 1];
            int[,] solution = new int[W + 1, weight.Length + 1];

            for (int i = 1; i <= W; i++)
            {
                for (int j = 1; j <= weight.Length; j++)
                {
                    d[i, j] = d[i, j - 1];
                    if (weight[j - 1] <= i)
                    {
                        int newValue = d[i - weight[j - 1], j - 1] + cell[j - 1];
                        if (newValue > d[i, j])
                        {
                            d[i, j] = newValue;
                            solution[i, j] = 1;
                        }
                    }
                }
            }

            int[] reconstructedSolution = new int[weight.Length];
            int W_remaining = W;
            for (int j = weight.Length; j > 0; j--)
            {
                if (solution[W_remaining, j] == 1)
                {
                    reconstructedSolution[j - 1] = 1;
                    W_remaining -= weight[j - 1];
                }
            }

            return (d[W, weight.Length], reconstructedSolution);
        }

        int withRep = KnapsackWithReps(W, weight, cell);
        var withoutRep = KnapsackWithoutReps(W, weight, cell);

        return (withRep, withoutRep);
    }
}
```

Консоль отладки Microsoft Visual Studio

С повторениями: 48
Без повторений: 46
Реконструированное решение: [1, 0, 1, 0]

I:\ИВТ-6-о-22-1\Алгоритмизация\11\ConsoleApp2\bin\Debug\net6.0\...
Чтобы автоматически закрывать консоль при остановке отладки, добавьте в файл "Properties\DebugSymbols.props" элемент "AutoCloseConsole" со значением "true".
Нажмите любую клавишу, чтобы закрыть это окно:

Нахождения длины НВП в списке

```
1 using System;
2 using System.Collections.Generic;
3
4 class Program
5 {
6     static void Main()
7     {
8         List<int> a = new List<int> { 7, 3, 1, 1, 8, 4, 9, 1, 2, 6, 5, 9, 3, 8, 1 };
9
10        var result = ListBottomUp2(a);
11
12        Console.WriteLine($"Длина самой длинной возрастающей подпоследовательности: {result.Item1}");
13
14        Console.WriteLine("Использование предыдущего списка: ");
15        Console.WriteLine($"{string.Join(" ", result.Item2.Item1)}");
16
17        Console.WriteLine("Без использования предыдущего списка: ");
18        Console.WriteLine($"{string.Join(" ", result.Item2.Item2)}");
19    }
20
21    static (int, (List<int>, List<int>)) ListBottomUp2(List<int> a)
22    {
23        List<int> d = new List<int>();
24        List<int> prev = new List<int>();
25
26        for (int i = 0; i < a.Count; i++)
27        {
28            d.Add(1);
29            prev.Add(-1);
30            for (int j = 0; j < i; j++)
31            {
32                if (a[j] < a[i] && d[j] + 1 > d[i])
33                {
34                    d[i] = d[j] + 1;
35                    prev[i] = j;
36                }
37            }
38        }
39
40        int ans = 0, maxIndex = 0;
41        for (int i = 0; i < d.Count; i++)
42        {
43            if (ans < d[i])
44            {
45                ans = d[i];
46                maxIndex = i;
47            }
48        }
49
50        List<int> listUsingPrev = RestoreUsingPrev(prev, maxIndex);
51        List<int> listWithoutPrev = RestoreWithoutPrev(ans, maxIndex, d, a);
52
53        return (ans, (listUsingPrev, listWithoutPrev));
54    }
55
56    static List<int> RestoreUsingPrev(List<int> prev, int maxIndex)
57    {
58        List<int> result = new List<int>();
59        while (true)
60        {
61            result.Add(maxIndex);
62            if (prev[maxIndex] == -1)
63                break;
64            maxIndex = prev[maxIndex];
65        }
66        result.Reverse();
67        return result;
68    }
69
70    static List<int> RestoreWithoutPrev(int ans, int maxIndex, List<int> d, List<int> a)
71    {
72        List<int> result = new List<int>();
73        while (true)
74        {
75            result.Add(maxIndex);
76            if (ans == 1)
77                break;
78            ans--;
79
80            while (true)
81            {
82                maxIndex--;
83                if (maxIndex < 0) break; // Добавлено условие для предотвращения выхода за границы массива
84                if (d[maxIndex] == ans && a[maxIndex] < a[result[result.Count - 1]])
85                    break;
86            }
87        }
88        result.Reverse();
89        return result;
90    }
91 }
92
```

Консоль отладки Microsoft Visual Studio

Длина самой длинной возрастающей подпоследовательности: 5
Использование предыдущего списка: [1, 3, 5, 9, 11]
Без использования предыдущего списка: [2, 3, 5, 10, 11]

I:\ИВТ-6-о-22-1\Алгоритмизация\11\ConsoleApp3\bin\Debug\net6.0\Con

Чтобы автоматически закрывать консоль при остановке отладки, включите опцию "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно: █