МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Информационные системы»

**Лабораторная работа №2**по дисциплине «Основы алгоритмизации и программирования»

Выполнил:  
студент(ка) гр. ПИбд-12 Дозорова Алена

Ульяновск  
2021

Задание на лабораторную работу:

1. Реализовать задачу  используя жадный алгоритм:

Вариант 6: реализовать задачу о рюкзаке

1. Реализовать задачу используя динамическое программирование:

Вариант 6: реализовать задачу о рюкзаке

1. Реализовать 3 алгоритма с разной сложностью

Краткое описание используемых алгоритмов и структур

В программе используется класс BackpackTask, который реализует задачу о рюкзаке, заполняя два массива - цен и весов – случайными значениями и выводит ответ двумя способами – через динамическое программирование и через жадный алгоритм (с использованием массива и через сортировку и заполнение максимальными значениями)

Три реализованных алгоритма: заполнение двумерного массива случайными символами (О(n^2)), бинарный поиск(логарифмическая сложность), поразрядная сортировка (O(n+k)).

Код лабораторной работы:

public class BackpackTask

{

public int[] price;

public int[] weigth;

int Weight;

public BackpackTask(int n, int w)

{

Weight = w;

Random r = new Random();

price = new int[n];

weigth = new int[n];

for (int i = 0; i < n; i++)

{

price[i] = r.Next(Weight + 1)+1;

weigth[i] = r.Next(Weight + 100)+1;

}

}

public void Print()

{

Console.WriteLine("\nЦены на предметы");

for (int i = 0; i < price.Length; i++)

{

Console.Write(price[i] + " ");

}

Console.WriteLine("\nМассы предметов");

for (int i = 0; i < weigth.Length; i++)

{

Console.Write(weigth[i] + " ");

}

}

//Необходимо подобрать такой набор вещей, чтобы он помещался в рюкзаке и имел максимальную ценность (стоимость).

public void knapsack\_din()

{

int[] mas = new int[Weight + 1];

mas[0] = 0;

for (int w = 1; w <= Weight; w++)

{

mas[w] = mas[w - 1];

for (int i = 0; i < weigth.Length; i++)

{

if (weigth[i] <= w)

{

mas[w] = Math.Max(mas[w], mas[w - weigth[i]] + price[i]);

}

}

}

Console.WriteLine("\nМаксимальная цена рюкзака " + mas[Weight]);

}

public void knapsack\_greedy()

{

int[] knapsack = new int [Weight+100];

for(int i = 0; i<weigth.Length; i++)

{

knapsack[weigth[i] / price[i]] = i;

}

Array.Sort(knapsack);

Array.Reverse(knapsack);

int cost = 0;

int mass = 0;

foreach (int i in knapsack)

{

if(mass+weigth[knapsack[i]]<= Weight)

{

mass += weigth[knapsack[i]];

cost = cost + price[knapsack[i]];

}

}

int max\_c = 1;

for (int i = 0; i<price.Length; i++)

{

max\_c = Math.Max(max\_c, price[i]);

}

if(max\_c == 1)

{

max\_c = 0;

}

int result = Math.Max(cost, max\_c);

Console.WriteLine("\nМаксимальная цена рюкзака " + result);

}

}

static void Main(string[] args)

{

int n, W;

Console.WriteLine("Введите количество предметов в рюкзаке и максимальный вес");

n = Convert.ToInt32(Console.ReadLine());

W = Convert.ToInt32(Console.ReadLine());

BackpackTask task = new BackpackTask(n, W);

task.Print();

task.knapsack\_din();

task.knapsack\_greedy();

sq\_array();

string[] a = { "aaabfh", "srrryhf", "dhhui" };

Console.WriteLine(BinarySearch(task.price, 45, 0, task.price.Length - 1));

Console.WriteLine("Было: "+a[0]+" "+a[1]+" "+a[2]);

a = RadixSort(a, 3, 3, 3);

Console.WriteLine("Стало: " + a[0] + " " + a[1] + " " + a[2]);

Console.ReadKey();

}

public static void sq\_array()

{

char[,] array = new char[5, 5];

for (int i = 0; i < 5; i++)

{

for (int j = 0; j < 5; j++)

{

array[i, j] = (char)i;

Console.Write(array[i, j] + " ");

}

Console.Write("\n");

}

}

static int BinarySearch(int[] array, int searchedValue, int left, int right)

{

while (left <= right)

{

var middle = (left + right) / 2;

if (searchedValue == array[middle])

{

return middle;

}

else if (searchedValue < array[middle])

{

right = middle - 1;

}

else

{

left = middle + 1;

}

}

return -1;

}

public static string[] RadixSort(string[] a, int k, int n, int m)

{

for (int i = 0; i < k; i++)

{

int[] c = new int[123];

string[] b = new string[n];

for (int j = 97; j < 123; j++)

{

c[j] = 0;

}

for (int j = 0; j < n; j++)

{

char d = a[j][m - i - 1];

c[d]++;

}

int cou = 0;

for (int j = 97; j < 123; j++)

{

int temp = c[j];

c[j] = cou;

cou += temp;

}

for (int j = 0; j < n; j++)

{

int d = a[j][m - i - 1];

b[c[d]] = a[j];

c[d]++;

}

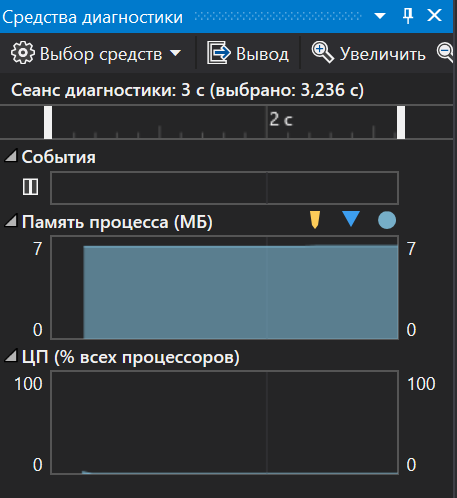
a = b;

}

return a;

}

Результаты тестирования:



Выводы:

Я разобралась с реализациями динамических и жадных алгоритмов, со сложностью алгоритмов.