

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”  
КАФЕДРА ИИТ

**ОТЧЁТ**  
по лабораторной работе №6  
**«Задача о рюкзаке»**

Выполнил:

Студент 2 курса  
группы ПО-9  
Харитонович Захар Сергеевич

Проверила:

Глущенко Т. А.

Брест 2022

### **Задание.**

1. Динамическим программированием решить классическую задачу о рюкзаке. Выбор входных параметров провести самостоятельно.
2. Письменно подробно ответить на вопросы.

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

```
using namespace std;
```

```
void findItems(int i, int j, vector<int> &result, vector<vector<int>> sum,
int *w) {
    if (sum[i][j] == 0) return;
    if (sum[i][j] == sum[i-1][j]) {
        findItems(i - 1, j, result, sum, w);
    } else {
        findItems(i - 1, j - w[i], result, sum, w);
        result.push_back(i);
    }
}
```

```
int main() {
    int capacity, n;
    cin >> capacity >> n;
    int weight[n + 1], cost[n + 1];
    for (int i = 1; i <= n; i++) {
        cin >> weight[i] >> cost[i];
    }
    vector<vector<int>> sum(n + 1, vector<int> (capacity + 1));
    for (int i = 0; i <= capacity; i++) {
        sum[0][i] = 0;
    }
    for (int i = 0; i <= n; i++) {
        sum[i][0] = 0;
    }
    for (int i = 1; i <= n; i++) {
        for (int j = 1; j <= capacity; j++) {
            if (j > weight[i]) {
                sum[i][j] = max(sum[i-1][j], cost[i] + sum[i-1][j -
weight[i]]);
            } else {
                sum[i][j] = sum[i-1][j];
            }
        }
    }
    vector<int> result;
    findItems(n, capacity, result, sum, weight);
    cout << "Items:" << endl;
    for (auto cur : result) {
        cout << cur << " ";
    }
    return 0;
}
```

### **Контрольные вопросы**

1. Какие точные методы решения данной задачи вы знаете?  
К точным алгоритмам относятся:  
полный перебор  
метод ветвей и границ  
динамического программирования (ДП)
2. Чему равна временная сложность решения данной задачи полным перебором?  
 $O(2^n)$
3. Чему равна временная сложность решения данной задачи динамическим программированием?

$O(n*m)$

4. Описать решение задачи о рюкзаке с приведенными ниже входными данными жадным алгоритмом и методом динамического программирования. Каким из методов мы находим оптимальное решение?

Грузоподъемность рюкзака,  $W = 4$ . Даны 3 предмета с весом  $w_i$  и ценой  $p_i$ :

$w_1=1, p_1=30$ ;  $w_2=2, p_2=50$ ;  $w_3=3, p_3=60$ ; 1

Жадным алгоритмом мы бы взяли первым делом  $w_1$  так как он самый дорогой и при добавлении любого другого предмета получили бы перевес. Методом динамического программирования мы получим оптимальное решение разбив данную задачу на подзадачи.

5. Что такое NP-полная задача и является ли задача о рюкзаке NP-полной задачей.

**NP** — класс задач, ответ на которые можно проверить за полиномиальное время.

Задача о рюкзаках является такой задачей.