

Практическая работа №16

Вариант №6 – Реализовать задачу о рюкзаке методом ветвей и границ.

Автор: Николаев-Аксенов И. С.

Группа: ИКБО-20-19

Код программы:

```
1. #include <iostream>
2. #include <algorithm>
3. #include <queue>
4. using namespace std;
5.
6. struct Item
7. {
8.     float weight;
9.     int value;
10. };
11.
12. struct Node
13. {
14.     int level, profit, bound;
15.     float weight;
16. };
17.
18. bool cmp(Item a, Item b)
19. {
20.     double r1 = (double) a.value / a.weight;
21.     double r2 = (double) b.value / b.weight;
22.     return r1 > r2;
23. }
24.
25. int bound(Node u, int n, int knapsackWeight, Item arr[])
26. {
27.     if (u.weight >= knapsackWeight)
28.         return 0;
29.
30.     int profit_bound = u.profit;
31.
32.     int j = u.level + 1;
33.     int totalWeight = u.weight;
34.
35.     while ((j < n) && (totalWeight + arr[j].weight <= knapsackWeight))
36.     {
37.         totalWeight += arr[j].weight;
38.         profit_bound += arr[j].value;
39.         j++;
40.     }
41.
42.     if (j < n)
43.         profit_bound += (knapsackWeight - totalWeight) * arr[j].value / arr[j].weight;
44.
45.     return profit_bound;
46. }
47.
48. int knapsack(int W, Item arr[], int n)
49. {
50.     sort(arr, arr + n, cmp);
51.
52.     queue<Node> Q;
53.     Node u, v;
54.
55.     u.level = -1;
56.     u.profit = u.weight = 0;
57.     Q.push(u);
```

```

58.
59.     int maxProfit = 0;
60.     while (!Q.empty())
61.     {
62.         u = Q.front();
63.         Q.pop();
64.
65.         if (u.level == -1)
66.             v.level = 0;
67.
68.         if (u.level == n - 1)
69.             continue;
70.
71.         v.level = u.level + 1;
72.         v.weight = u.weight + arr[v.level].weight;
73.         v.profit = u.profit + arr[v.level].value;
74.
75.         if (v.weight <= W && v.profit > maxProfit)
76.             maxProfit = v.profit;
77.
78.         v.bound = bound(v, n, W, arr);
79.
80.         if (v.bound > maxProfit)
81.             Q.push(v);
82.
83.         v.weight = u.weight;
84.         v.profit = u.profit;
85.         v.bound = bound(v, n, W, arr);
86.         if (v.bound > maxProfit)
87.             Q.push(v);
88.     }
89.
90.     return maxProfit;
91. }
92.
93. int main() {
94.     setlocale(LC_ALL, "Russian");
95.
96.     int W = 10;
97.     Item arr[] = {{2, 40}, {3.14, 50}, {1.98, 120}, {5, 95}, {3, 30}};
98.     int n = sizeof(arr) / sizeof(arr[0]);
99.
100.    cout << "Максимальная стоимость: " << knapsack(W, arr, n);
101.
102.    return 0;
103. }

```

Результат выполнения программы:

Максимальная стоимость: 255