

Knapsack

Algoritmi Avasați

Buhai Darius - 234

1 Problema 1

1.1 A

```
1 int max_sum(int k, const vector<int>& nums){
2     int total = 0;
3     unordered_set<int> values;
4     values.insert(0);
5     for(auto num: nums)
6         for(auto val: values)
7             if(val + num <= k){
8                 total = max(total, val + num);
9                 values.insert(val + num);
10            }
11     return total;
12 }
```

1.2 B

```
1 int max_sum(istream &in){
2     int k, x, total = 0;
3     in>>k;
4     while(in>>x){
5         if(x + total <= k)
6             total += x;
7         else if(total < x)
8             total = x;
9     }
10    return total;
11 }
```

Pentru a ajunge la o soluție cel puțin $\frac{1}{2}$ OPT vom aduna elemente din șir fără a depăși valoarea k . Astfel, se vor crea 2 cazuri:

1. Cazul în care suma noastră finală ajunge $\geq \frac{k}{2}$, condiția noastră fiind îndeplinită.
2. Cazul în care suma noastră ajunge $\leq \frac{k}{2}$, următorul element din listă (x_i) fiind $\geq k - \frac{k}{2}$, respectiv $x_i \geq \frac{k}{2}$, caz în care suma totală este înlocuită de acest element (x_i) iar condiția este din nou satisfăcută.