Knapsack

Given:

- weights $w[i]$, $i = \overline{1,n}$
- values $c[i]$, $i = \overline{1,n}$
- W - maximum wight, capacity of teh knapsack

Find subset of items I such that:

- The total weight of items in I does not exceed W, $\sum_{i \in I} w[i] \leq W$
- The total vlue of items in I is maximized W, $max_{I \subset \{1,\ldots,n\}} \sum_{i \in I} c[i]$

Base knapsack solution

- $dp[i][]$ - stores the minimal achievable weight for the first i elements with values c.
- $0 \leq i \leq n, 0 \leq c \leq C$
- dp recalc:

$$dp[i][c] = \begin{cases} dp[i-1][c], & \text{если } c[i] > c, \\ \min\left(dp[i-1][c], dp[i-1][c - c[i]] + w[i]\right), & \text{иначе.} \end{cases}$$

final asymptotics: $\underline{O}(n * C_{max})$

Approximation polynomial knapsack

Given:

- weights $w[i]$, $i = \overline{1,n}$
- values $c[i]$, $i = \overline{1,n}$
- W - maximum weights, capacity of the knapsack
- ε - acceptable error.

Find subset of items I such that:

- The total weight of items in I does not exceed W, $\sum_{i \in I} w[i] \leq W$
- Maximum the total vlue of items in I differs from the real answer by no more than $1 + \varepsilon$
- The algorithm works in polynomial asymptotics.

Approximated polynomial solution

1. Find $K = \frac{\varepsilon * C_{max}}{n}$, $C_{max} = max_{i=\overline{1,n}} c[i]$

2. $c^{'}[i] = \lfloor \frac{c[i]}{K} \rfloor$

3. Solve based knapsack problem with $c^{'}[i]$

final asymptotics: $\underline{O}(n^2 \lfloor \frac{C_{max}}{K} \rfloor) = \underline{O}(n^2 \lfloor \frac{n}{\varepsilon} \rfloor)$
Correct: Proof of the correctness of this algorithm.