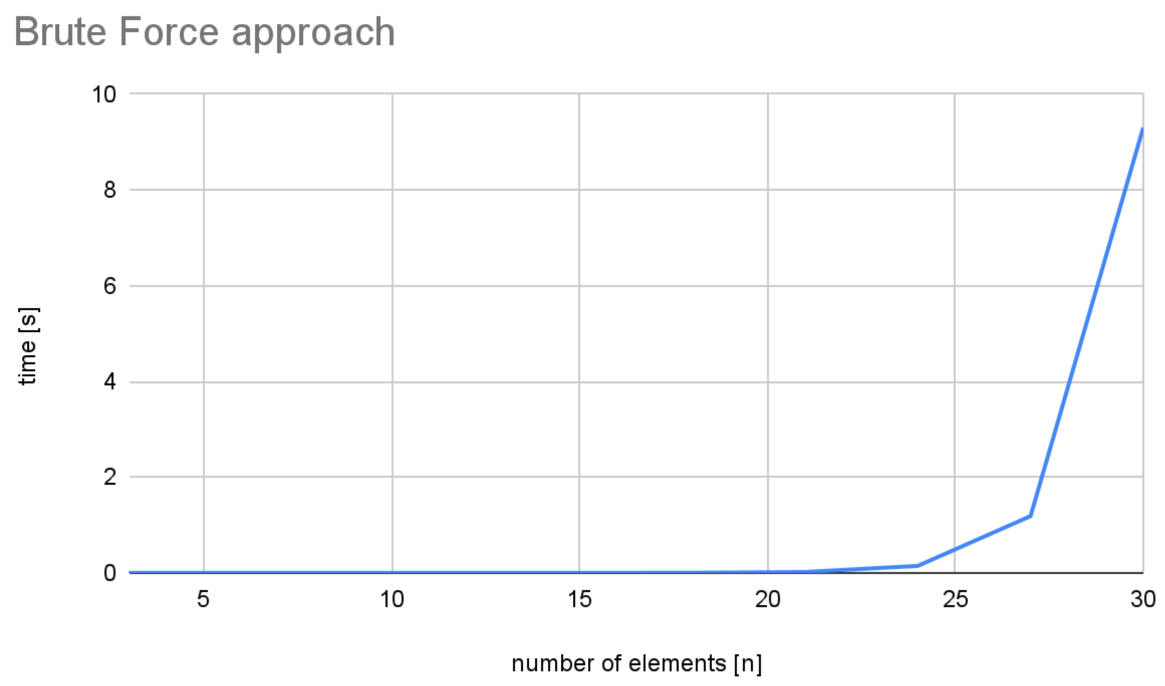
Małgorzata Gierdewicz 148264

Knapsack Problem

Knapsack problem is an optimization problem, where we decide which items should go to a backpack with given capacity.

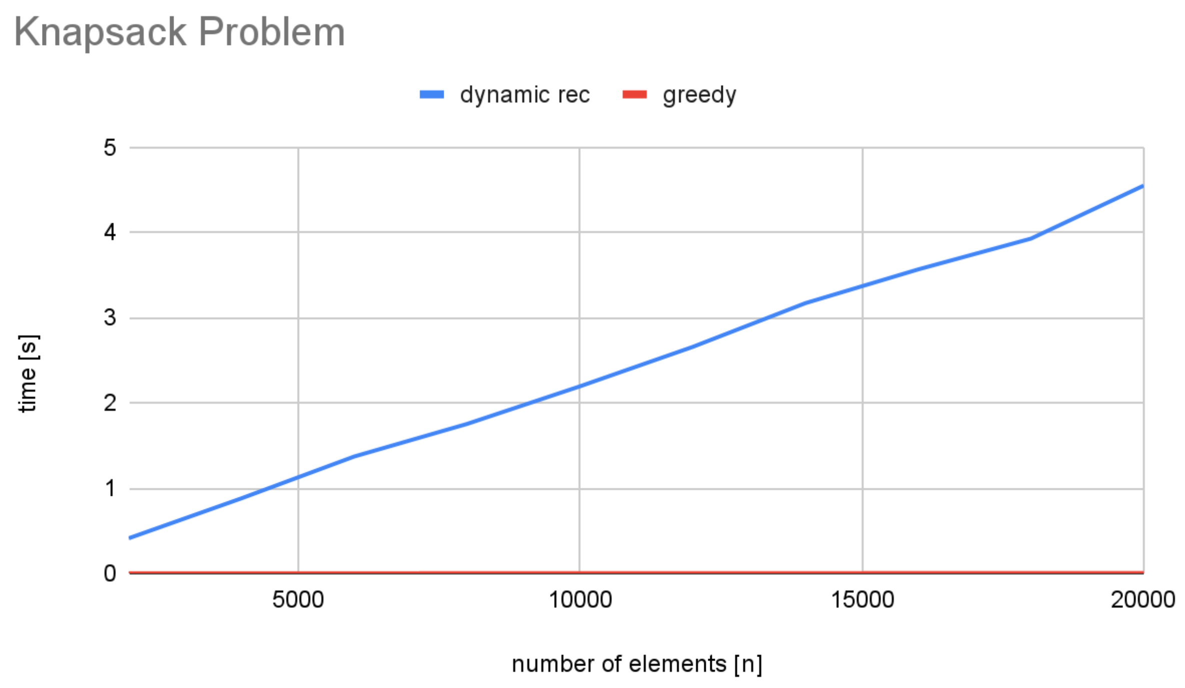
Brute force approach is the most intuitive approach. We check every item to decide whether it can fit into the set of optimal solutions, and then we check whether other combinations of items are possible. Unfortunately, this approach is very time- inefficient, since we check a set of permutations, so the time complexity of brute force approach is O(2^n). The only upside to that is that no matter what capacity of the backpack is, it won't influence the complexity of the algorithm (it is capacity- insensitive).



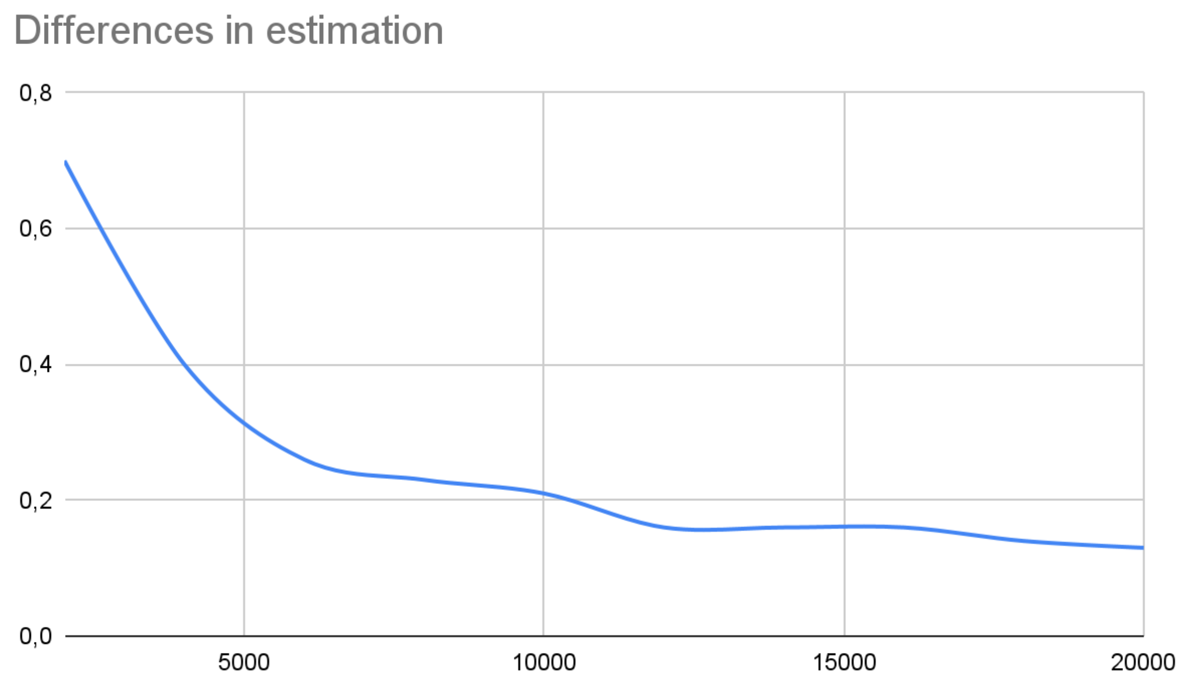
On the graph below, three other approaches are presented:

- dynamic programming
- greedy approach

Dynamic Programming allows us to store solutions to already computed problems in 2D array, so that the same instance won't be computed many times, such as in brute force approach. The time complexity is polynomial and is equal to O(n*m), where n = number of elements and m = maximum capacity. I



The other approach used is the greedy approach, which is the most time efficient out of all three approaches, but it is not precise. That is because we don't know if by using greedy-like approach we find the local or global optimum. The following chart represents the difference in estimation of set intervals for optimal solutions between upper and lower bound greedy algorithm. Increasing number of elements in problem increases the algorithm accuracy.


Differences in estimation