

580: Algorithms  
Tutorial: Dynamic Programming

1. A thief can carry  $k$  kilograms of loot in his *knapsack*. He robs a shop containing  $N$  items. Item  $i$  is worth  $b_i$  bitcoin and weighs  $k_i$  kilos. The thief wants to decide which items to take to maximise the total value he steals.
  - (a) How would you decompose this problem into subproblems? Does the problem have optimal substructure and overlapping subproblems?
  - (b) Write an algorithm that, given an array  $B$  such that  $B[i]$  is the value of item  $i$  and an array  $K$  such that  $K[i]$  is the weight of item  $i$ , and a maximum weight  $k$ , solves the problem in  $\mathcal{O}(kN)$  time.
  - (c) Since he is greedy, the thief attempts to use the following strategy: the next item chosen should be the one with the maximum value per weight ratio. The items are sorted in descending order of this ratio, and the thief takes items in this order until no more items can be taken. Does this greedy strategy always find the optimal solution?

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu\_assist\_pro