

Assignment 7

1. Consider the maximal value interval selection problem from class: we are given a list of intervals $I_1 = (s_1, e_1, v_1), I_2 = (s_2, e_2, v_2), \dots, I_n = (s_n, e_n, v_n)$ where s_i, e_i , and v_i are the start and end times and value of interval I_i . Our goal is to select a collection of non-overlapping intervals of maximal value.

In class a proposed algorithm that I realized later was actually incorrect was to sort the intervals by value per total time, and then pick intervals greedily according to that strategy.

Convince me that this algorithm is incorrect.

2. Consider the knapsack problem from class: we are given a list of items $e_1 = (w_1, v_1), e_2 = (w_2, v_2), \dots, e_n = (w_n, v_n)$ and a maximal weight W . Consider the algorithm that orders the items by value per unit weight and greedily picks according to this ordering. Argue that this algorithm is incorrect.

3.
 - (a) How many digits are required to write the number 1?
 - (b) How many digits are required to write the number 1000?
 - (c) Determine an upperbound to the number of digits to write a number n .
 - (d) In class we gave a $O(nW)$ algorithm to solve the knapsack problem. Think about part c and ask yourself if this algorithm is *technically* a polynomial time algorithm - we'll see later in this semester that it's very unlikely that there is *strongly* polynomial time solution. https://en.wikipedia.org/wiki/Time_complexity#Strongly_and_weakly_polynomial_time