

Lab03-Greedy Strategy

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2021.

* If there is any problem, please contact TA Haolin Zhou.

* Name:_____ Student ID:_____ Email: _____

1. *Interval Scheduling*. Interval Scheduling is a classic problem solved by **greedy algorithm**: given n jobs and the j -th job starts at s_j and finishes at f_j . Two jobs are compatible if they do not overlap. The goal is to find maximum subset of mutually compatible jobs. Tim wants to solve it by sort the jobs in descending order of s_j . Is this attempt correct? Prove the correctness of such idea, or else provide a counter-example.
2. *Done deal*. In a basketball league, teams need to complete player trades through matching contracts. Every player is offered a contract. For the sake of simplicity, we assume that the unit is M , and the size of all contracts are integers. The process of contract matching refers to the equation: $\sum_{i \in A} a_i = \sum_{j \in B} b_j$, where a_i refers to the contract value of player i in team A involved in the trade and b_j refers to the value of player j in team B .
Assume that you are a manager of a basketball team and you want to get **one** star player from another team through trade. The contract of the star player is $n(n \in \mathbb{N}^+)$. The goal is to complete the trade with as few players as possible.
 - (a) Describe a **greedy** algorithm to get the deal done with the least players in your team. Assume that there are only 4 types of contracts in your team: $25M$, $10M$, $5M$, $1M$, and there is no limit to the number of players. Prove that your algorithm yields an optimal solution.
 - (b) Suppose that the available contract sizes are powers of c , i.e., the values are c^0, c^1, \dots, c^k for some integers $c > 1$ and $k \geq 1$. Show that the greedy algorithm always yields an optimal solution.
 - (c) Give a set of contract sizes for which the greedy algorithm does not yield an optimal solution. Your set should include a $1M$ so that there is a solution for every value of n .
3. *Set Cover*. **Set Cover** is a typical kind of problems that can be solved by greedy strategy. One version is that: Given n points on a straight line, denoted as $\{x_i\}_{i=1}^n$, and we intend to use minimum number of closed intervals with fixed length k to cover these n points.
 - (a) Please design an algorithm based on **greedy** strategy to solve the above problem, in the form of *pseudo code*. Then please analyze its *worst-case* complexity.
 - (b) Please prove the correctness of your algorithm.
 - (c) Please complete the provided source code by C/C++ ([The source code *Code-SetCover.cpp* is attached on the course webpage](#)), and please write down the output result by testing the following inputs:
 - i. the number of points $n = 7$;
 - ii. the coordinates of points $x = \{1, 2, 3, 4, 5, 6, -2\}$;
 - iii. the length of intervals $k = 3$.

Remark: Screenshots of running results are also acceptable

Remark: You need to include your .pdf and .tex files in your uploaded .rar or .zip file.