

Lab03-GreedyStrategy

CS214-Algorithm and Complexity, Xiaofeng Gao, Spring 2020.

* If there is any problem, please contact TA Shuodian Yu.

* Name: ShaoxiongLin Student ID: 517010910028 Email: Johnson-Lin@sjtu.edu.cn

1. There are $n + 1$ people, each with two attributes $(a_i, b_i), i \in [0, n]$ and $a_i > 1$. The i -th person can get money worth $c_i = \frac{\prod_{j=0}^{i-1} a_j}{b_i}$. We do not want anyone to get too much. Thus, please design a strategy to sort people from 1 to n , such that the maximum earned money $c_{max} = \max_{1 \leq i \leq n} c_i$ is minimized. (Note: the 0-th person doesn't enroll in the sorting process, but a_0 always works for each c_i .)
 - (a) Please design an algorithm based on greedy strategy to solve the above problem. (Write a pseudocode)
 - (b) Prove your algorithm is optimal.

Solution. (a)

Algorithm 1: *greedy1*

Input: $n + 1$ attribute pairs (a_i, b_i) to be sorted in some way,
 i ranges from 0 to n
Output: the minimized maximum earned money c_{max}

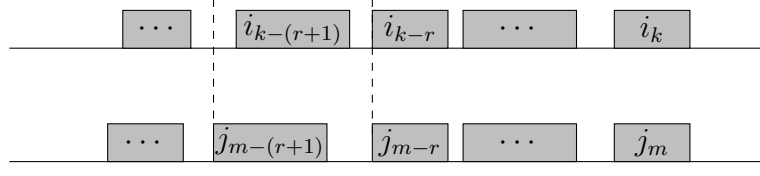
```
1 sort the  $(a_i, b_i)$  by  $a_i \times b_i$  in nondecreasing way except  $(a_0, b_0)$ ;  
2  $c_{max} \leftarrow 0, prod \leftarrow 1$ ;  
3 for  $i \leftarrow 1$  to  $n$  do  
4    $prod \leftarrow prod \times a_{i-1}$ ;  
5    $c_i \leftarrow \frac{prod}{b_i}$ ;  
6   if  $c_i > c_{max}$  then  
7      $c_{max} \leftarrow c_i$ ;  
8   else  
9     continue;  
10 return  $c_{max}$ ;
```

- (b) For the sake of convenience, given an arrangement of people, we say the arrangement has an inversion if there are two people, say i and j , such that $a_i \times b_i < a_j \times b_j$ but j stands before i in the sequence. Define S to be the solution provided by the Alg.1, it is easy to see S has no inversion. Define S^* to be an optimal arrangement that has the fewest number of inversions. If S^* has no inversion, then $S^* = S$. If S^* has an inversion, let i, j be an adjacent inversion. Swapping i and j does not affect the money earned by the people before as well as people behind them, but since $a_i \times b_i < a_j \times b_j$, c_j used to be $\frac{\prod_{k=0}^{j-1} a_k}{b_j}$, c_i used to be $\frac{\prod_{k=0}^{i-1} a_k \times a_j}{b_i}$, after swapping, a'_j now is $\frac{\prod_{k=0}^{j-1} a_k \times a_i}{b_j}$ and a'_i now is $\frac{\prod_{k=0}^{i-1} a_k}{b_i}$. It is easy to see $\max(a_i, a_j) > \max(a'_i, a'_j)$, thus after swapping we have $c_{max} \leq c'_{max}$, it is still an optimal solution.

□

2. **Interval Scheduling** is a classic problem solved by greedy algorithm and we have introduced it in the lecture: 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.

Proof. This attempt is correct. Assume this attempt is not optimal. Let i_1, i_2, \dots, i_k denote set of jobs selected by this attempt. Let j_1, j_2, \dots, j_m denote set of jobs in an optimal solution with $i_k = j_m, i_{k-1} = j_{m-1}, \dots, i_{k-r} = j_{m-r}$ for the largest possible value of r . We can replace $j_{m-(r+1)}$ with $i_{k-(r+1)}$, solution still feasible and optimal, but contradicts the maximality of r .



□

3. There are n lectures numbered from 1 to n . Lecture i has duration (course length) t_i and will close on d_i -th day. That is, you could take lecture i **continuously** for t_i days and must finish before or on the d_i -th day. The goal is to find the maximal number of courses that can be taken. (Note: you will start learning at the 1-st day.)

Please design an algorithm based on greedy strategy to solve it. You could use the data structure learned on Data Structure course. You need to write pseudo code and prove its correctness.

Solution. Given a schedule S , we say it has an inversion if there is a pair of lectures i and j such that $d_j > d_i$ or $d_j = d_i, t_j > t_i$ but j scheduled before i .

Suppose S is the solution provided by Alg.2, and S^* is an optimal solution with inversions. We can say that S will not have inversion, and we can change S^* step by step into S without hurting its quality. Let i and j be an adjacent inversion, swapping i and j does not affect the lectures before as well as after them, and since $d_j > d_i$ or $d_j = d_i, t_j > t_i$, after swapping, the number of courses that can be taken will not reduce. We can do this repeatedly until there is no inversion in S^* , at that time S^* is actually S .

□

Algorithm 2: *greedy2*

Input: n pairs (d_i, t_i) to be sorted in some way, i ranges from 1 to n

Output: the set of the maximal number of courses that can be taken

```

1 sort the  $(d_i, t_i)$  by  $d_i$  in nondecreasing way, sort by  $t_i$  in
  nondecreasing way(not broken the result of  $d_i$ );
2  $A \leftarrow \emptyset$ ;
3 for  $i \leftarrow 1$  to  $n$  do
4   if  $job_i$  is compatible with  $A$  then
5      $A \leftarrow A \cup \{job_i\}$ ;
6   else
7     continue;
8 return  $A$ ;
```

4. Let S_1, S_2, \dots, S_n be a partition of S and k_1, k_2, \dots, k_n be positive integers. Let $\mathcal{I} = \{I : I \subseteq S, |I \cap S_i| \leq k_i \text{ for all } 1 \leq i \leq n\}$. Prove that $\mathcal{M} = (S, \mathcal{I})$ is a matroid.

Proof. First, we prove that \mathcal{I} is hereditary. Suppose $X \subseteq Y$ and $Y \in \mathcal{I}$, that is $Y \subseteq S$ and $|Y \cap S_i| \leq k_i$ for all $1 \leq i \leq n$. Since $X \subseteq Y$, we have $X \subseteq S$ and $|X \cap S_i| \leq |Y \cap S_i| \leq k_i$ for all $1 \leq i \leq n$, so $X \in \mathcal{I}$, thus \mathcal{I} is hereditary.

Second, we prove that \mathcal{M} satisfies the exchange property. Suppose $A \in \mathcal{I}, B \in \mathcal{I}$, and $|A| < |B|$, then there must be an element, say x , such that $x \in B$ but $x \notin A$. Since S_1, S_2, \dots, S_n is a partition of S , then there must be a set, say S_j , such that $x \in B \cap S_j$. We construct a new set $A \cup \{x\}$, it easy to see that $A \cup \{x\} \subseteq B \subseteq S$ and for any $i \neq j$, we have $A \cup \{x\} \cap S_i = A \cap S_i$, thus $|A \cup \{x\} \cap S_i| = |A \cap S_i| \leq k_i$. If $i = j$, we have $|A \cup \{x\} \cap S_i| \leq |B \cap S_i| \leq k_i$. Thus, $A \cup \{x\} \in \mathcal{I}$, so \mathcal{M} satisfies the exchange property. Finally, we can say that $\mathcal{M} = (S, \mathcal{I})$ is a matroid. \square