University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, M. Fuchs, A. Malyusz

Algorithm Theory
Sample Solution Exercise Sheet 9

Due: Monday, 20th of November 2024, 10:00 am

Assumption: You may assume that calculations with real numbers can be performed with arbitrary precision in constant time.

Exercise 1: Unlock Achievement *(8 Points)*

There are N skills numbered 1 to N and M achievements numbered 1 to M.
Each skill has a positive integer level, and the initial level of every skill is 1.
You can pay a cost of C_i yen to increase the level of skill i by 1. You can do this as many times as you want.
Achievement i is achieved when the following condition is satisfied for every $j = 1, \ldots, N$, for which you will receive a reward of A_i yen.
Condition: The level of skill j is at least $L_{i,j}$.
Find the maximum possible total reward obtained minus the total cost required when appropriately choosing how to raise the skill levels in $O((NL + M)(NL + NM)^2)$ running time, where L is the maximum level a skill can have.

Hint: $\max(\text{Reward\_earned} - \text{Money\_Spent}) = -\min(\text{Money\_Spent} - \text{Reward\_earned}) = \text{All\_rewards} - \min(\text{Money\_Spent} + \text{Reward\_missed})$. A cut defines a partition of the skills into bought/not bought and the skills into achieved/not achieved.

Sample Solution

We will create a min cut problem. The value of the cut will have help us give the optimal value and the cut defines which levels are bought and which achievements are reached. More precisely in the min cut if a skill node is with T it means we have bought it. If it is an achievement we reached it. If the node is with S it means we have missed/unsatisfied it. Let us create the min cut problem in the following way:
Every skill level will have a node associated with it $S_{i,l}$ where i is the skill number and l is the level of the skill.
Every achievement will have a node associated with it A_j where i is the achievement number.
Additionally, we have a source node S and a sink node T.
Then we will add an edge from S to $S_{i,l}$ with capacity C_i and an edge from $S_{i,l}$ to T with capacity 0.
We will add an edge from S to A_j with capacity 0 and an edge from A_j to S with capacity D_j.
Then we will add an edge from $S_{i,l}$ to $S_{i,l-1}$ with capacity ∞. I can not buy a level when a level lower than it has not been bought.
Then we will add an edge from $S_{i,L_{j,i}}$ to A_j with capacity ∞. An achievement can only be claimed if the skill's level is bought.

Exercise 2: Round Robin *(7 Points)*

There are N players numbered 1 through N participating in a round-robin tournament. Specifically:

- For every pair (i, j) where $1 \leq i < j \leq N$, Player i and Player j play a match against each other exactly once.
- This results in a total of
$$\binom{N}{2} = \frac{N(N-1)}{2}$$
matches.
- In every match, one player is declared the winner, and the other is the loser; there are no draws.

Match Results

- M matches have already been completed.
- In the i-th completed match, Player W_i won against Player L_i.

Objective

Determine all players who can still become the **unique winner** of the tournament after all remaining matches are played. A player is considered the unique winner if:

- The number of matches they win is strictly greater than the number of matches won by any other player.

Solve it in $O(N^5)$ running time.

Hint: Try to solve the question of "Is ith player the unique winner?". Try to create nodes for matches too not only for the players.

Sample Solution

For every person we will answer the question of can they become the unique winner? For this we will create a max flow problem.
Assume that Player p will win any players against which the result of the match has not been determined. Let win be the number of Player p's wins. If $win = 0$, then he cannot be the unique winner, so hereinafter we assume $win \geq 1$.
Prepare vertices S, T, $A_{i,j}$ $(1 \leq i < j \leq N)$, and B_i $(1 \leq i \leq N)$. S is the source, T is the sink, $A_{i,j}$ represents the match between Players i and j, and B_i represents Player i.
Add the following edges with capacities 1:

- An edge $S \to A_{i,j}$
- For every match between Players i and j:
  - If i won, add an edge $A_{i,j} \to B_i$
  - If j won, add an edge $A_{i,j} \to B_j$
  - If they have not had a match yet, add an edge $A_{i,j} \to B_i$ and an edge $A_{i,j} \to B_j$
- Add win number of edges $B_p \to T$
- Add $(win - 1)$ number of edges B_i $(i \neq p) \to T$

The maximum flow from S to T on this graph is $\frac{N(N-1)}{2}$ if and only if Player p may be the unique winner.
The capacities of the edges on this graph are all 1, and the number of edges is $O(N^2)$, so it can be solved in an $O(N^3)$ time.

Exercise 3: Min of a minimum cut *(5 Points)*

You are given a flow network, i.e., a directed graph $D = (V, E)$ a source $s \in V$ and a sink $t \in V$, and two capacity functions on the edges $c_1, c_2 : E \to \mathbb{N}$. Your task is to find an $s-t$ cut in polynomial time that is minimum with respect to c_1 and among such cuts, it is minimum with respect to c_2. Prove that your algorithm is correct.

Hint: Try to create a new capacity function and prove that using the max flow on it will give the desired result.

Sample Solution

Let us use a new capacity function $c = (|E| + 1) \cdot (\max_{e \in A}\{c_2(e)\})c_1 + c_2$. Let us denote $B := (\max_{e \in E}\{c_2(e)\})$. In the following we show that a cut is minimum cut according to c if and only if it is minimum according to c_1 and according to these minimum for c_2:
$\Rightarrow$: Contradiction suppose, that this D_1 cut is not optimal for c_1. So there exists a cut D_2 for which $c_1(D_2)$ is less. c_2 has at most $|E| \cdot B$ capacity in the D_1 cut and 0 at least in D_2. This means the following: $c(D_1) \leq (|E| + 1) \cdot B \cdot c_1(D_1) + (|E|) \cdot B \leq (|E| + 1) \cdot B \cdot c_1(D_2)$ (the last one is because D_1 is optimal for c), and we know that $c_1(D_1) \geq c_1(D_2)$. If reorder the terms to one side we can see that it is a negative number even though it should be a non-negative one.
$\Leftarrow$: Same way.

Exercise 4: Problem for the exercise session *(0 Points)*

We have a grid with N horizontal rows and N vertical columns. Let (i, j) denote the square at the i-th row from the top and j-th column from the left. A character $c_{i,j}$ describes the color of the square (i, j). The character B means the square is painted black, W means the square is painted white, and ? means the square is not yet painted.
Takahashi will complete the black-and-white grid by painting each unpainted square black or white. Let the *zebraness* of the grid be the number of pairs of a black square and a white square sharing a side.
Find the maximum possible zebraness of the grid that Takahashi can achieve.