Tsinghua University

Discrete Optimistic

Fall 2024

Homework 1

Lin Zejin February 26, 2025

- **Collaborators:** I finish this homework by myself.

Problem 1. By using contraction view, it suffices to show that the first step can actually be viewed as using blue rules sequentially.

For each node n_i, assume that v_i is the edge connected to n_i of minimal weight.

WLOG, we assume $\omega(n_i), 1 \leq i \geq |V|$ is increasing.

Then we prove that coloring v_i blue simultaneously is equivalent to applying blue rule to the cut $(\{n_1, \cdots, n_k\}, \{n_{k+1}, \cdots, n_{|V|}\})$ successively.

By induction. For applying blue rule to the cut $(\{n_1\}, \{n_2, \cdots, n_{|V|}\})$, v_1 turns blue since v_1 is of the minimal weight in all edges.

If we have colored $v_1, \cdots, v_{k-1}$, then for applying blue rule to the cut $(\{n_1, \cdots, n_k\}, \{n_{k+1}, \cdots, n_{|V|}\})$, v_k turns blue. Otherwise, if $v'_k = (a, b)$ turns blue, $a \in \{n_1, \cdots, n_k\}$ and $b = n_t \in \{n_{k+1}, \cdots, n_{|V|}\}$, then $\omega(v'_k) \geq \omega(v_t)$ by the primitive rule. Thus $\omega(v'_k) \geq \omega(v_t) \geq \omega(v_k)$ as $t \geq k$, which causes contradiction.

So actually we can apply blue rule one by one. Therefore, *Borůvka* algorithm is indeed applying blue rule constantly, and it terminates until there is only one blue component, *i.e.* forming a blue tree. As we proved in class, this tree is MST.

Problem 2. WLOG, we assume that $J_1, \cdots, J_n$ is exactly a possible schedule of minimal time cost.

If $f_i \leq f_{i+1}$, then swap J_i, J_{i+1}. It will not affect other jobs. However, the time when job J_i, J_{i+1} finished both will be changed. It is $\max\{t + p_{i+1} + p_i + f_i, t + p_{i+1} + f_{i+1}\}$, which is not longer than $\max\{t + p_i + p_{i+1} + f_{i+1}, t + p_i + f_i\} = t + p_i + p_{i+1} + f_{i+1}$ at first. So it is still a possible schedule of minimal completion time cost.

Using bumbling sort algorithm, we can rearrange $J_1, \cdots, J_n$ such that f_i is decreasing, and it is still a possible schedule of minimal completion time cost.

Since the schedule of minimal time cost must exist, it suffices to sort f_i to get a possible schedule of minimal completion time cost.

Problem 3.

$$\sum_{i=1}^{n} \omega_i C_i = \sum_{i=1}^{n} \omega_i \left(\sum_{j=1}^{i} t_j \right) = (\sum_{i=1}^{n} t_i)(\sum_{i=1}^{n} \omega_i) - \sum_{i=1}^{n} \omega_i \left(\sum_{j=i+1}^{n} t_j \right) \tag{3.1}$$

So it suffices to maximize $\sum_{i=1}^{n} \omega_i \left(\sum_{j=i+1}^{n} t_j \right)$.

WLOG, we assume that $(t_i, \omega_i), 1 \leq i \leq n$ has been a possible solution.

If $\omega_i \left(\sum_{j=i+1}^{n} t_j \right) < \omega_k \left(t_i - t_k + \sum_{j=i+2}^{n} t_j \right)$, $i < k$ then swap (ω_i, t_i) and (ω_k, t_k). It only affects $i \sim k$-th terms in the RHS of (3.1). Noticed that the difference

$$
\begin{aligned}
&\sum_{t=i}^{k} \omega_t \left(\sum_{j=t+1}^{n} t_j \right) - \left[\sum_{t=i+1}^{k-1} \omega_t \left(\sum_{j=t+1}^{n} t_j - t_k + t_i \right) + \omega_k \left(\sum_{j=i+1}^{n} t_j - t_k + t_i \right) + \omega_i \left(\sum_{j=k+1}^{n} t_j \right) \right] \\
&= \sum_{t=i+1}^{k-1} \omega_t (t_k - t_i) + \omega_i \sum_{j=i+1}^{k} t_j - \omega_k \sum_{j=i}^{k-1} t_j \\
&= \omega_i \left(\sum_{j=i+1}^{n} t_j \right) - \omega_{i+1} \left(t_i + \sum_{j=i+2}^{n} t_j \right) \\
&> 0
\end{aligned}
$$

So (3.1) will be smaller as they swap, which contradicts that (3.1) has already taken the minimal value.

So a possible solution of minimal value must satisfy that ω_i

Problem 4. The problem can be reduced to the original case.

We can enumerate each job. Take the first job, and remove other jobs that are not compatible with the first job. Here we remove the interval of the first job, and hence get a line of jobs. Apply the original algorithm of interval scheduling to the rest of jobs and we get a number that maximize the numbers of compatible jobs in the case that the first job is taken. Compare such n numbers we can obtain the answer.

Problem 5.

Problem 6.

Problem 7. (a) Since all edge costs are distinct, applying blue rule or red rule is the unique operation at any time. Therefore, the final graph with a blue tree is unique. Since MST contains all blue edges as we prove in the lecture, MST has to be the unique blue tree.

(b) Yes, here is the proof.