Greedy Algorithm with Haoyu

Jiawei Li

June 2024

1 Introduction

Notations. A forest F of n vertex are given. Each vertex i takes t_i time to resolve, and the weight of that vertex is w_i. Assume $w_i, t_i \geq 0$ for any $i \in [n]$.

The goal is to an optimal order $a_1, \ldots, a_n$ of visiting n vertices that minimize the sum of cost of all vertices (cost is defined below).

The greedy algorithm.

Definition 1 (available chunk)**.** For any forest F, an available chunk C is a subset of vertices in F such that

1. the vertices in C are connected.
2. the root of C is also a root in F.

Definition 2 (Score Function)**.** For any subset C, the score function $s(C)$ is $w(C)/t(C)$, where $w(C) \coloneqq \sum_{x \in C} w_x, t(C) \coloneqq \sum_{x \in C} t_x$.

The greedy algorithm A will proceeds as follow: In each step,

1. A calculates the score function of all available chunk.
2. Let C be the available chunk with highest score; if there are multiple choices, pick C to be the smallest one. If there are still choices, break-tie arbitrarily.
3. A then visits the root of C.

The sequence generated by the greedy algorithm A has the following property.

Lemma 3. *Let C be the available chunk pick by A in any step, then A will visit all the vertices in C in the next $|C|$ steps.*

Proof. We prove by induction on n, i.e., the size of the remaining forest. When $n = 1$, this is trivially true. Now suppose this is true for any forest of size at most $n - 1$.

After querying the root r_C of C, C is divided into several connected sub-chunk $C_1, \ldots, C_m$. If $m = 0$, this is trivial true. Now suppose $m \geq 1$.

We will show that the next $|C| - 1$ steps of A can be described by m stages: At the start of each stage, one of the remaining C_i will be the chunk picked by A, and then A will *completely* visit every vertex in C_i in the next $|C_i|$ steps in this stage.

By the induction hypothesis, in any steps of A, if one of the C_i is the chunk picked by A, A will *completely* visit every vertex in C_i in the next $|C_i|$ steps. Thus, we only need to rule out the possibility that at the start of some stage, there exists a chunk C^* that is better than any remaining C_i.

To this end, we prove two useful claims here:

Claim 4. $s(C_i) > s(C)$ *for any* $i \in [m]$.

Proof. Prove this claim by contradiction: if this is not true for C_i, then consider the set $C' = C \backslash C_i$. Since $s(C_i) \leq s(C)$ and $s(C)$ is the weighted average of $s(C')$ and $s(C_i)$, we must have $s(C') \geq s(C)$. Moreover, we have $|C'| < C$. Thus, this is contradict to the behavior of A. ◇

Claim 5. *A will not query any nodes that are not in the sub-tree of r_C before finishing querying every node in $C_1, \ldots, C_m$.*

Proof. Any chunk not in the sub-tree of r_C has strictly worse score function compared to $C_1, \ldots, C_m$ by the previous claim. $\diamond$

By the previous claim, there are two remaining possibilities:

1. C^* shares its root with one of remaining C_i. This is impossible, because replacing C_i with C^* in the original chunk C would make C better.
2. C^* has no overlap with any one of the remaining C_i. This is also impossible because replacing C with $C \cup C^*$ would make C better.

$\square$

Observation 6. *If an order is optimal, then every suffix of it is also optimal w.r.t. the sub-problem of removing previously visited vertices.*

Proof. Easy to prove by contradiction. $\square$

Theorem 7. *The greedy algorithm A is optimal.*

Proof. Prove by induction on n. When $n = 1$, this is trivially true. Now suppose the greedy algorithm A is optimal for any forest of size at most $n-1$. Let $a = (a_1, \ldots, a_n)$ be the order generated by A.

Now take any other order $b = (b_1, \ldots, b_n)$. If $b_1 = a_1$, by the induction hypothesis, $(a_2, \ldots, a_n)$ is the optimal order for the sub-problem without $a_1 (= b_1)$. So order b cannot be better than order a.

Now assume $b_1 \neq a_1$. Let C be the first chunk picked by A and assume $b_j = a_1$. We first generate a new order b' such that $b'[1, \ldots, j-1]$ is same as b, but we then run algorithm A on the sub-problem without $b_1, \ldots, b_{j-1}$, which generate the rest of sequence b'. By the induction hypothesis, b' is no worse than b.

Now assume $b'_k = a_1$ $(k \geq j > 1)$. Note that every vertex in C has yet been visited in the first $k-1$ steps of b'. We claim that C must be the chunk picked by A in the sub-problem where $b'[1, \ldots, k-1]$ are removed. If not, assume A pick C', then C' should also be the chunk picked by A at the start, contradict to the definition of C.

Therefore, by Lemma 3, we have that the sub-sequence $a[1, \ldots, |C|]$ is equal to the sub-sequence $b'[k, \ldots, k+|C|-1]$. Let's consider swap $b'[1, \ldots, k-1]$ with $b'[k, \ldots, k+|C|-1]$ in the order b'. Denote the new order by b''. Since the score function of C is no worse than that of $\{b'_1, \ldots, b_{k-1}\}$, (by some standard calculation), b'' is no worse than b'. Now b'' has the same initial vertex as a, so a is no worse than b''. Combining everything, we have that a is no worse than b.

$\square$