Solution de Pizza Delivery

Hugo

5 juin 2019

Il s'agit d'un problème polonais : `https://szkopul.edu.pl/problemset/problem/q_HBwDECevrQ2iQh1wT6ssx2/site/?key=statement`. On s'intéresse ici directement à la dernière sous-tâche (problème complet).

1 Résumé du problème

On considère un arbre à N nœuds, enraciné en 1, dont les arêtes sont **pondérées**. La pondération d'une arête correspond au temps mis pour la traverser.

On veut visiter tous les nœuds en un minimum de temps, sachant qu'on a le droit de se téléporter directement à la racine au plus $K-1$ fois (éteindre le chauffage = se téléporter à la racine)

Contraintes : $1 \leq N, K \leq 10^5$. Les poids c_i sont des entiers entre 1 et 10^6.

2 Analyse du cas $K = 1$: suivre un DFS

Évidemment, supposer $K = 1$ enlève la notion de téléportation, on doit juste trouver le chemin le plus court passant par tous les nœuds.

Dans ce cas, la solution optimale est bien sûr de suivre le chemin d'un DFS depuis la racine. La longueur est $2\sum c_i$ (le double de la somme des poids des arêtes), vu que chaque arête sera parcourue deux fois (pour descendre dans un sous-arbre, puis une fois l'exploration finie pour remonter au parent). Il est évidemment minimal.

3 Augmentation de K et propriétés des téléportations

Cela veut dire qu'on va à certains moments dans notre parcours, on va se téléporter à la racine. On peut s'intéresser aux nœuds depuis lesquels on effectue ces téléportations. En usant de l'intuition et/ou en jouant avec des exemples, on peut faire une succession de remarques :

Observation 1 : Lorsqu'on se téléporte depuis un noeud u dans une solution optimale, u était le dernier noeud découvert avant la téléportation.

Preuve : Soit d le dernier noeud découvert avant la téléportation. Si $d \neq u$, on s'est déplacé de d vers u sans découvrir de nouveaux nœuds avant de revenir à la racine, ce qui est une perte de temps. Il aurait été plus optimal de se téléporter depuis d directement.

Corollaire : On ne se téléporte jamais deux fois depuis le même noeud.

Observation 2 : Dans une solution optimale, on ne se téléporte jamais depuis un noeud qui n'est pas une feuille.

Preuve : Supposons qu'on se téléporte depuis un noeud $u \neq 1$ qui n'est pas une feuille. Vu que u est le dernier noeud découvert à ce moment-là (selon l'observation 1), les enfants de u ne sont pas encore découverts. Cela veut dire qu'on devra redescendre dans le sous-arbre de u plus tard. Il aurait donc été plus optimal de se téléporter depuis le parent de u.

4 Vers une caractérisation d'une solution optimale

Les observations précédentes vont nous permettre d'user d'un **point méthode** très puissant : caractériser une solution optimale, dont la forme originelle est complexe et donc difficilement manipulable (chemin cassé par des téléportations), par un choix à la forme plus simple.

Il s'agit d'une technique qu'on doit toujours tenter d'utiliser lorsqu'on tente de résoudre un problème où la solution a une forme complexe, difficile à "saisir" mentalement.

Il s'agit ici de caractériser une solution par les feuilles depuis lesquels on se téléporte. On colorie en noir ces feuilles ("trous noirs") et en blanc les autres.

On caractérise donc ici une solution optimale par l'ensemble des feuilles noires (depuis lesquelles on décide de se téléporter). Pour que ce soit une vraie caractérisation (une sorte de bijection), il faut qu'on puisse retrouver la solution elle-même à partir du sous-ensemble de feuilles choisi.

5 Retour du subset noir à la solution optimale

On s'est ici ramené à trouver un sous-ensemble de feuilles S optimal. Pour que cette réduction soit pertinente, il faut **entre autres** qu'on puisse déduire efficacement du choix de S la solution optimale.

Supposons donc qu'on fixe l'ensemble des nœuds noirs, sur lesquels on sait qu'on se téléportera (on ne sait pas dans quel ordre cependant).

On va encore se baser sur un DFS, mais on voit qu'intuitivement qu'il vaut mieux découvrir un maximum de choses avant de se "rapprocher" d'une téléportation, ce qui évitera parfois de revenir pour finir de découvrir les autres branches.

Posons un peu de vocabulaire pour formaliser cela : on définit la "couleur" c_u d'un noeud u comme le nombre de feuilles noires contenues dans son sous-arbre. Un noeud est dit incolore si sa couleur est 0.

On peut reprendre le principe général dans les observations 1 et 2, qui est que "laisser des trucs non découverts alors qu'on aurait pu les découvrir sans frais, c'est pas très cool".

Lorsqu'on rentre dans un sous-arbre, il vaut donc mieux commencer par explorer les enfants incolores, avant d'explorer les enfants colorés desquels on ne remontera pas.

Quel est le coût de cette solution ? On peut calculer pour chaque noeud $u \neq 1$, le nombre de fois d_u où on parcourt l'arête qui descend vers u :

- Si $c_u = 0$ (aucune feuille noire), alors on est obligé de descendre et de remonter du sous-arbre, donc $d_u = 2$
- Sinon, $c_u > 0$, on va donc descendre une fois pour chaque téléportation. Pas plus, car on aura bien vu toutes les feuilles blanches avant de se téléporter depuis la dernière feuille noire (car on explore en priorité les nœuds incolores). On aura jamais besoin de remonter. On a donc $d_u = c_u$.

On vérifie aisément que pour un certain S imposé, d_u est minimal pour chaque u, ce qui est condition suffisante *(mais pas nécessaire !)* pour montrer que notre solution est optimale.

Conclusion de cette partie : Cette partie est "nécessaire" dans le sens où il faut qu'on puisse revenir du choix du S optimal à la solution au problème optimale. Mais elle fait en réalité bien plus que ça. On exprime ici la valeur de la solution (longueur du chemin cassé) dérivée du choix de S **en fonction de S uniquement**, ici des valeurs c_u ! Cela permet de vraiment caractériser ce qu'est un sous-ensemble de feuilles optimal...

6 Trouver le subset noir optimal

On reprend les notations précédentes en rajoutant p_u le poids de l'arête qui descend vers u. La valeur d'un sous-ensemble S sera donc $V = \sum p_u \cdot d_u$, où $d_u = 2, 1, 2, 3, 4, \ldots$ pour $c_u = 0, 1, 2, 3, 4, \ldots$.

Ce qui est intéressant dans cette expression, c'est qu'elle fait apparaître les variations de V lorsqu'on rajoute une feuille dans S (ajouter 1 à c_a pour tout ancêtre a de la feuille).

Plus exactement : pour les ancêtres incolores ($c_a = 0$), on retire p_a à V (ça minimise donc on dit "valeur ajoutée positive") et pour les ancêtres déjà colorés, on rajoute p_a à V (on dit "valeur ajoutée négative", car ça nous arrange pas).

Cela nous suggère d'étudier l'impact des ancêtres sur les coûts des feuilles. Initialement, la "valeur ajoutée" de toute feuille est la somme des poids p_a de ses ancêtres. Dès qu'on rajoute une feuille u, **certains** ancêtres vont se colorer pour la première fois et leur impact va passer en négatif.

Pour chacun de ces **certains** ancêtres, qui forment un préfixe des ancêtres de u, on doit ajouter $-2p_a$ aux valeurs ajoutées de toutes les feuilles de a. Cela se fait évidemment par ajout sur intervalle avec un segment tree en lazy propagation, lorsque les feuilles sont triées par date de découverte du DFS (on maintient pour chaque noeud l'intervalle de ses feuilles).

Cela nous suggère le glouton suivant : on part de $S = \emptyset$, de valeur initiale $2 \cdot \sum p_u$. Tant qu'il y'a une feuille dont la valeur ajoutée est positive (et que $|S| < K$), on ajoute la feuille de valeur ajoutée maximale dans S.

La correction de ce glouton n'est pas évidente. Un moyen efficace de l'approcher serait ici de tenter de construire des contre-exemples avec des stratégies variées, ce qui peut faire apparaître des éléments de preuve suffisamment convaincants.

7 Preuve de correction du glouton (argument d'échange)

On considère notre solution gloutonne $G = g_1, g_2, \ldots$ et une solution optimale $O = o_1, o_2, \ldots$ comme des séquences **ordonnées**, finies, d'ajouts, avec $O \neq G$. Soit $N(O)$ l'indice de la première différence entre G et O. Supposons par l'absurde que O maximise $N(O)$ (parmi toutes les solutions optimales), et donc que $G \neq O$.

Soit u la feuille ajoutée dans G au moment de cette première différence. Si u était ajouté dans O plus tard, on aurait pu l'ajouter dès maintenant sans faire varier la valeur de la solution.

Dans le cas contraire : rappelons que u a une valeur ajoutée strictement positive a ce moment-là. Considérons tous les nœuds v_i dont la valeur ajoutée est réduite par l'ajout de u à ce moment-là. On ignore les v_i à la fois présent dans O et dans G. S'il n'y aucun v_i dans O après u, ça ne coûte rien de rajouter u dans O à ce moment-là.

Sinon, voici l'idée-clé de la preuve : considérons p le v_i dans O (après u) qui soit "**le plus proche** de u" en terme de LCA. La valeur ajoutée de p était plus faible que u à ce moment-là, sinon G n'aurait pas choisi u. On peut supposer sans perte de généralité que p soit le noeud ajouté dans O au moment de la première différence (quitte à échanger deux nœuds ce qui change rien). On observe qu'ajouter p ferait au moins toutes les descentes de valeur ajoutée causées par u (car p est le plus proche en terme de LCA). Pourtant, p a une valeur ajoutée plus faible. On peut donc remplacer p par u en gardant l'optimalité.

Dans tous les cas, on arrive à construire une nouvelle solution optimale O' vérifiant $N(O') > N(O)$, ce qui est contradictoire avec l'hypothèse faite. Notre solution G est donc optimale.

8 Conclusion

Nous avons résolu le problème avec un algorithme glouton s'appuyant sur la structure "segtree avec lazy". Complexité : $O(N \log N)$.

Code : `https://pastebin.com/B4caYrAm`

Que retenir de ce problème ? Ce qui le rend intéressant est cet alignement d'observations. On pourrait croire que ce soit de la pure intuition. Pourtant cet alignement ne sort pas de nulle part : il se base sur un fil conducteur tenu par une pensée méthodique. La méthode est ici ce qui permet d'avancer, lentement mais sûrement, dans la recherche des observations requises.

— Savoir extraire les bénéfices de l'intuition grâce à la rigueur. Dégager clairement des "principes vagues" (par exemple, explorer un max avant de se téléporter), tenter de les clarifier au maximum, les formaliser si c'est pertinent. Il est vraiment indispensable de ne rien laisser de "pas très clair", même si on a la flemme de plonger dans les détails. Il faut cependant ne pas avoir peur de s'appuyer temporairement sur des principes vagues.

— Noter un maximum d'observations, être exhaustif. Il faut vraiment rien laisser passer, même si on n'est pour le moment pas convaincu que ce soit vrai. Ne pas avoir peur des gloutons, mais toujours rester méfiant évidemment.

— Pour tenter de prouver (ou juste de tester) quelque chose, il peut être très utile de chercher à construire des contre-exemples variés. Si la propriété est vraie, l'échec de construction de contre-exemples pourra dans certains cas suggérer les éléments de preuve.

— A côté de chaque "pensée", mettre un "statut" (à quel point on pense que c'est vrai, utile + est-ce qu'on l'a exploité). Pour être efficace, on jongle souvent entre les réflexions et il faut bien maintenir l'état de chacune.

— Être sur d'exploiter **totalement** les remarques faites. Ne pas hésiter à les formuler de différentes manières, de chercher des corollaires ou au contraire des généralisations...