DM9 (éléments de réponse)

Ce corrigé, fourni par le jury du concours de l'école Polytechnique, ajoute quelques commentaires en direction des candidats.

Question 1. On se place dans le graphe dont les sommets sont les entiers de 1 à n et les arêtes sont les couples (k, p_k). Par hypothèse, tout sommet y est de degré sortant 1 et les seuls cycles du graphe sont des boucles, c'est-à-dire des cycles de longueur 1. Puisque le chemin obtenu en partant d'un entier m et en remontant la liste de ses ancêtres se termine par un cycle, il contient nécessairement une boucle, centrée sur un chef de m.
En particulier, ce chef est un ancêtre de tous les ancêtres de m, donc m ne peut pas avoir deux chefs, qui seraient ancêtres l'un de l'autre. En outre, deux entiers k et ℓ ont un ancêtre commun si et seulement s'ils ont le même chef, qui sera le chef de cet ancêtre commun.
Cette question a pour but de mettre le candidat en confiance et de lui permettre de s'approprier la représentation de l'algorithme Union-Find qu'il devra manipuler au cours de l'épreuve. Elle vise également à évaluer sa capacité à introduire des graphes de manière pertinente, puis à raisonner rigoureusement sur ces graphes.

Question 2. Voici un tableau où l'on a recensé, après chaque étape de l'algorithme, le parent et le poids de chaque nœud ; la situation initiale est indiquée à l'étape 0.

Étape	p_1	w_1	p_2	w_2	p_3	w_3	p_4	w_4
0	1	0	2	0	3	0	4	0
1	1	1	1	0	3	0	4	0
2	1	1	1	0	3	1	3	0
3	1	2	1	0	1	1	3	0
4	1	1	1	0	1	1	1	0

La seule difficulté est ici de ne pas oublier que la requête $\mathrm{Test}(2,4)$ *entraîne une compression du chemin liant le sommet* 4 *à son chef* 1*, qui devient désormais son parent. Cette question vise donc à s'assurer que les élèves ont bien retenu que compresser les chemins dans l'algorithme Union-Find était indispensable ou, le cas échéant, à le leur rappeler.*

Question 3. Au cours de l'algorithme, et tant qu'il est chef, un entier peut d'abord voir son poids augmenter (de 1 à chaque fois), au cours de fusions à l'occasion desquelles on lui assigne de nouveaux descendants ; puis, si un jour il cesse d'être chef, son poids ne changera plus jamais. Par conséquent, si un entier v est un jour le parent d'un entier u, soit w_v et w_u leurs poids à ce moment-là. Puisque u n'est plus chef, il ne le sera plus jamais, donc $w_u^\infty = w_u$; par ailleurs, le poids de v ne pourra qu'augmenter, donc $w_v^\infty \geqslant w_v > w_u = w_u^\infty$.
Cette question est la première question délicate du sujet. Elle a pour but de vérifier si le candidat maitrise la dynamique des liens de parenté et des poids au cours de l'algorithme, et s'il est capable de jouer à la fois sur l'état du système à un moment de l'exécution de l'algorithme et une fois cette exécution achevée. Par ailleurs, cette question et chacune des questions suivantes a pour but de préparer les questions ultérieures, notamment le résultat final que l'on souhaite démontrer.

Question 4. On note Γ le graphe étiqueté dont les sommets sont les entiers de 1 à n et les arêtes sont les couples (u, v) pour lesquels $u \neq v$ et v a été parent de u au cours de l'algorithme ; chaque sommet k est étiqueté par l'entier w_k^∞. On dit alors que v est un Γ-descendant de u (qui est un Γ-ancêtre de v) si Γ contient un chemin allant de u à v.
Chaque entier k gagne ses ancêtres un par un, au cours de requêtes Union ; chaque nouvel ancêtre de k devenant un ancêtre de tous les anciens ancêtres de k. Par conséquent, Γ contient un chemin partant de k et passant par tous ses Γ-ancêtres, dont les poids ne font qu'augmenter le long du chemin. En particulier, deux sommets de même étiquette n'ont aucun Γ-descendant commun.
Or, une récurrence sur w indique que, à tout moment au cours de l'algorithme, chaque sommet de poids $w \geqslant 0$ possède au moins 2^w descendants. Les s entiers k pour lesquels $w_k^\infty = \ell$, ces s entiers ont donc, à eux tous, un total d'au moins $2^\ell s$ ou plus Γ-descendants.
En outre, tout sommet qui Γ-descend d'un entier de poids non nul a dû être un des arguments d'une requête Union. Il y a eu au plus m requêtes Union, donc au plus $2m$ entiers utilisés comme arguments de telles requêtes. On en conclut que $s \geqslant 2m/2^\ell$.
Dans la continuité de la question précédente, cette question a pour but d'affiner la compréhension de la dynamique du système étudié par le candidat. Sa difficulté principale consiste à introduire une structure de données simple mais qui contiendra des informations valides à différentes phases de l'algorithme. Elle vise également à s'assurer que le candidat est à même de naviguer entre des échelles locale et globale, ce genre d'approches étant crucial en informatique.

Question 5. En question précédente, les s entiers k pour lesquels $w_k^\infty = \ell$ ont cumulé au plus $\min\{n, 2m\}$ Γ-descendants, ce qui nous fournit l'inégalité $s \leqslant \min\{n, 2m\}/2^\ell$, légèrement meilleure que l'inégalité précédente. En particulier, le plus grand des entiers w_k^∞ est un entier ℓ pour lequel $1 \leqslant s \leqslant \min\{n, 2m\}/2^\ell$, de sorte que $\ell \leqslant \log_2(\min\{m, n\}) + 1$.
En outre, toute requête portant sur deux entiers dont les chefs sont de poids w et w' a une complexité $\mathcal{O}(w + w')$. On conclut en remarquant que w et w' sont majorés par $\mathcal{O}(\ell) \subseteq \mathcal{O}\left(\log_2(\min\{m, n\})\right)$.

Cette question a pour but de vérifier si le candidat a le réflexe d'affiner de lui-même la borne qui lui avait été proposée en question précédente, et s'il est capable d'utiliser des arguments simples pour trouver des bornes supérieures sur la complexité de telle opération.

Question 6. Considérons une requête Union ou Test d'arguments a et b, et soit ω_a et ω_b le nombre de liens de parentés que cette requête remonte en partant de a et de b. La complexité de cette requête est majorée par $\mathcal{O}(\omega_a + \omega_b + 1)$, et celle-ci résulte en l'introduction d'au moins $(\omega_1 - 1) + (\omega_b - 1)$ arêtes dans E : il s'agit des les liens de parentés détruits par notre requête, et ceux-ci ne seront jamais recréés. Ainsi, si notre requête résulte en l'ajout de e arêtes dans E, sa complexité est majorée par $\mathcal{O}(e+1)$. Puisque l'on a m requêtes en tout et que la somme des entiers e ainsi définis est égale à $|E|$, le résultat désiré s'ensuit.
De nouveau, cette question a pour but de vérifier que le candidat est capable d'estimer la complexité d'un algorithme en fonction de paramètres auxquels il n'est pas habitué.

Question 7. Soit u un élément de V_ℓ. Les sommets accessibles par une arête de $\mathcal{G}$ partant de u sont des Γ-ancêtres de u. Ils sont donc d'étiquettes différentes, et au plus $a_{\ell+1} - a_\ell$ d'entre eux appartiennent à V_ℓ. Or, le nombre de sommets de Γ d'étiquette w est un entier $s_w \leqslant m/2^{w-1}$. Le nombre total d'arêtes internes de $\mathcal{G}$ internes à V_ℓ est donc majoré par

$$a_{\ell+1}\,|V_\ell| \leqslant a_{\ell+1} \sum_{w \geqslant a_\ell} m/2^{w-1} = a_{\ell+1} m/2^{a_\ell - 2} = 4m$$

Par ailleurs, chaque requête d'arguments a et b résulte en l'ajout dans $\mathcal{G}$ d'au plus deux arêtes quittant V_ℓ : il s'agit, sur chacun des deux chemins allant de a et b à leurs chefs et que l'on explore avec cette requête, de la dernière arête partant d'un sommet de V_ℓ. Par conséquent, au plus $2m$ arêtes sortent de V_ℓ.
Il s'agit là de la question la plus difficile du sujet, censée résister même aux élèves les plus brillants.

Question 8. Puisque $w_k^\infty \leqslant \log_2(m) + 1$ pour tout entier k, on sait que $V_\ell = \varnothing$ dès lors que $a_\ell > \log_2(m) + 1$. Or, la fonction $\log^\star$ est croissante, et $\log^\star(a_\ell) = \ell$ pour tout entier $\ell \geqslant 1$. Ainsi, lorsque $m \geqslant 2$ et $\ell > \log^\star(m)$, on sait que $\log^\star(a_\ell) = \ell > \log^\star(m) \geqslant \log^\star\left(\log_2(m) + 1\right)$, de sorte que $a_\ell > \log_2(m) + 1$ et $V_\ell = \varnothing$.
En particulier, $\mathcal{G}$ contient au plus $6m$ arêtes issues de chacun des $\lfloor \log^\star(m) \rfloor + 1$ ensembles V_ℓ susceptibles d'être non vides, et aucune arête issue des autres ensembles V_ℓ, nécessairement vides. Il contient donc $\mathcal{O}(m \log^\star(m))$ arêtes, ce qui conclut.
Cette question vise non seulement à obtenir explicitement le résultat espéré depuis le début du sujet, mais aussi à vérifier que le candidat est à même de prendre du recul sur les calculs de plus en plus compliqués qu'il a dû mener au cours des questions précédentes. On pourrait s'étonner qu'elle soit substantiellement plus abordable que la question 7. Néanmoins, les candidats auraient tout à fait eu le droit de demander à traiter cette question en admettant la question 7, avant de revenir sur celle-ci ensuite. Dans ce genre de cas, il ne faut pas s'auto-censurer, et ne pas hésiter à proposer à l'examinateur pareille démarche ; ici, la réponse aurait été positive, même s'il faut bien sûr que le candidat soit prêt à renoncer à son idée si jamais l'examinateur lui répond de manière négative.