Université Lyon 1
Département Informatique

M1
2019/2020

Mif12
Algorithmique distribuée

TD2 - Éléments de correction

Solution. Note des rédactrices : certaines rédactions pourraient être améliorées, et tout n'est pas garanti sans erreur. □

Exercice 1 : Algorithmes de tri

Solution. *Objectif : Compléter ce qui n'a pas été vu en cours et comparer le nombre de messages échangés sur un exemple.* □

Q 1. Indiquez pour chaque algorithme de tri suivant s'il est asynchrone ou faiblement synchrone :
- algorithme simple centralisé
- tri pair - impair
- shear sort
- bucket sort

Solution. Voici les définitions mises en jeu : :
- Asynchrone : les noeuds avancent à leur rythme, les messages des uns et des +autres sont envoyés quand l'algorithme le nécessite.
- Faiblement synchrone : l'algorithme est découpé en ronde. Dans chaque ronde, chaque noeud attend un message de chacun de ses voisins, puis effectue des instructions locales en fonction, entre autres, de ce qu'il a reçu, puis envoie +un message à tous ses voisins
- Synchrone : les noeuds sont parfaitement synchronisés sur une même échelle de temps.

- algorithme simple centralisé : asynchrone
- tri pair - impair : asynchrone
- shear sort : asynchrone
- bucket sort : faiblement synchrone

□

Q 2. Un algorithme de tri nécessitant plus de messages qu'un autre algorithme de tri a-t-il forcément un temps d'exécution plus long ?

Solution. Non, on ne peut rien dire. Tout dépend du degré de parallélisation de l'algorithme distribué. □

Exercice 2 : Encore un algorithme distribué pour construire un ensemble indépendant maximal !

Solution. *Objectif : Faire quelques preuves théoriques sur un algorithme distribué...* □

Considérons un graphe dont le degré maximum est noté Δ et comprenant n nœ]uds. L'algorithme que vous allez étudier fonctionne en $log\Delta + 1$ phases, chaque phase ayant $O(\log n)$ rondes.

À chaque ronde de la i^e phase, chaque nœud, qui fait toujours partie de l'algorithme, se marque avec probabilité $\frac{2^i}{10.\Delta}$. Les nœuds, qui se sont marqués et qui n'ont aucun voisin qui s'est marqué, rejoignent l'ensemble indépendant maximal (noté EIM par la suite). Les nœuds qui viennent de rejoindre l'EIM et leurs voisins quittent l'algorithme. Si à n'importe quel moment, un nœud devient isolé (c'est-à-dire que ses voisins ne font plus partie de l'algorithme), alors il rejoint l'EIM [1].

Q 1. Montrez que l'ensemble des sommets qui ont rejoint l'EIM (et qui peut évoluer au cours de l'algorithme) dans une phase donnée est toujours un ensemble indépendant.

Solution. Supposons que 2 voisins u et v ont été ajoutés à l'EIM. Supposons que le nœud v ait été ajouté à la ronde t. Le nœud u n'a pas pu être ajouté à la même ronde d'après l'algorithme (seuls "les nœuds, qui se sont marqués et qui n'ont aucun voisin qui marqué, rejoignent l'ensemble indépendant maximal (noté EIM par la suite)."). Le nœud u n'a pas pu être ajouté lors d'une ronde suivant la ronde t, car d'après l'algorithme, u étant un voisin de v, il a été enlevé de l'algorithme lorsque v a rejoint l'EIM. On peut appliquer exactement le même raisonnement si u avait rejoint en 1er (avant v) l'EIM. Donc 2 voisins ne peuvent pas être ajoutés à l'EIM. Enfin, l'ajout de nœud isolé à l'EIM ne viole pas la notion d'ensemble indépendance de l'EIM. □

Le but des questions suivantes est de prouver par récurrence sur i que chaque nœud toujours présent dans l'algorithme à la fin de la phase i a un degré d'au plus $\frac{\Delta}{2^i}$ avec forte probabilité. *Aide : on utilisera le fait que dans les conditions de l'algorithme, on a* $(1 - \frac{2^i}{10.\Delta})^{2.\frac{\Delta}{2^{i-1}}} \geq 4^{-2/5} \geq \frac{1}{2}$

Solution. Pour prouver ce résultat, on peut poser $x = \frac{2^i}{\Delta}$ et étudier la fonction $f(x) = (1 - \frac{x}{10})^{4/x}$ pour $0 < x \leq 2$ avec Wolfram Alpha par exemple `https://www.wolframalpha.com/input/?i=%281-x%2F10%29%5E%284%2Fx%29+for+x+between+0,1+and+2`. La fonction est strictement décroissantsur cet intervalle (ouvert en 0) et elle tend vers $e^{-2/5}$ en 0, qui est environ 0.67, quantité elle même supérieure à $\frac{1}{2}$. □

Q 2. Montrez que la proposition est vraie pour $i = 0$, c'est-à-dire avant le tout début de l'algorithme.

Solution. Pour $i = 0$, c'est-à-dire avant le tout début de l'algorithme, c'est vrai, puisque le degré maximum du graphe est Δ. □

Supposons maintenant qu'on est au début de la i^e phase, et que les nœuds restants dans l'algorithme ont un degré d'au plus $\frac{\Delta}{2^{i-1}}$ (proposition vraie à la phase $i-1$ par induction).

Q 3. Quels sont les cas simples pour lesquels la proposition est vraie à la fin de la phase i ?

Solution. À chaque ronde de la phase i, on peut avoir les cas suivants pour un nœud v :
- v est enlevé de l'algorithme et donc on n'a rien à prouver car on ne s'intéresse qu'aux nœuds présents dans l'algorithme à la fin de la phase i...
- Au plus $\frac{\Delta}{2^i} - 1$ voisins de v restent dans l'algorithme, et auquel cas on a bien prouvé ce qu'on voulait.

□

Intéressons-nous au cas, où, à la fin d'une ronde de la phase i, au moins $\frac{\Delta}{2^i}$ voisins d'un nœud v restent dans l'algorithme.

Q 4. Montrez que pour ce cas, le nœud v sera retiré de l'algorithme à la ronde suivante avec une probabilité d'au moins $\frac{1}{20}$.

Solution. Évaluons la probabilité, qu'à la ronde suivante, un voisin u de v 1) soit marqué et 2) qu'aucun voisin de u et qu'aucun voisin de v (excepté u) soit marqué. La probabilité qu'un voisin u de v vérifie ces 2 propriétés est au moins $\frac{2^i}{10.\Delta}.(1 - \frac{2^i}{10.\Delta})^{2.\frac{\Delta}{2^{i-1}}}$.

Ici on applique l'hypothèse de récurrence sur les voisins de v, qui du coup ont forcément un degré supérieur à $\frac{\Delta}{2^{i-1}}$.

D'après l'aide, $\frac{2^i}{10.\Delta}.(1 - \frac{2^i}{10.\Delta})^{2.\frac{\Delta}{2^{i-1}}} \geq \frac{2^i}{10.\Delta}.\frac{1}{2}$.

1. Précision pour "faire partie de l'algorithme". Moralement, on peut imaginer que l'ensemble des nœuds du graphe initial est dans une "worklist" au début, et que si un nœud est évincé au cours de l'algorithme, cela signifie qu'il sort de cette worklist

De plus, on ne peut pas avoir 2 voisins de v qui vérifient, en même temps, ces 2 propriétés (sinon elles ne pourraient pas être vérifiées). On peut alors sommer les probabilités car elles sont indépendantes et disjointes (sur l'ensemble des voisins de v, ie au moins $\frac{\Delta}{2^i}$ dans le cas étudié). Donc la probabilité qu'au moins un des voisins de v vérifie ces 2 propriétés est au moins $\frac{\Delta}{2^i} \cdot \frac{2^i}{10.\Delta} \cdot \frac{1}{2} = \frac{1}{20}$.

Pour résumer : à chaque ronde de la phase i, si un nœud a, à la fin de cette ronde, un degré (correspondant au nombre de voisins restants) égal à $\frac{\Delta}{2^i}$, alors le nœud v sera retiré de l'algorithme à la ronde suivante avec une probabilité d'au moins $\frac{1}{20}$. □

Q 5. Supposons que $100 \log n$ rondes doivent être appliquées à chaque phase de l'algorithme. Quelle est la probabilité qu'un nœud reste dans l'algorithme avec un degré d'au moins $\frac{\Delta}{2^i}$ à la fin de la phase i ? Cette probabilité sera exprimée en fonction de n. Conclure sur la preuve par induction.

Solution. D'après la question précédente, la probabilité pour qu'un nœud reste, à la fin de la phase i, avec un degré d'au moins $\frac{\Delta}{2^i}$ est au plus $(1 - \frac{1}{20})^{100. \log n}$. Or, $(1 - \frac{1}{20})^{100. \log n} \leq 2^{-5. \log n} = \frac{1}{n^5}$. Donc, à la fin de la phase i, la probabilité pour qu'un nœud reste dans l'algorithme avec un degré d'au plus $\frac{\Delta}{2^i}$ est au moins $1 - \frac{1}{n^5}$.

pour montrer l'équation du milieu, on prend le log en base 10 des deux cotés, cela donne $100log(n)log(19/20) \leq ?-5log(n)log(2)$ **et cela fait disparaître la dépendance sur** n**, ensuite un petit coup de calculette...** □

Q 6. Quelle est la probabilité pour qu'au moins un nœud reste, à la fin de la phase i, avec un degré d'au moins $\frac{\Delta}{2^i}$?

Solution. On a une proba sur un noeud donné, maintenant on veut "au moins un noeud".

En appliquant l'inégalité de Boole (proba de l'union inférieure à la somme de sprobas), la probabilité pour qu'au moins un nœud reste, à la fin de la phase i, avec un degré d'au moins $\frac{\Delta}{2^i}$ est inférieure ou égale à n fois la probabilité qu'un nœud reste, à la fin de la phase i, avec un degré d'au moins $\frac{\Delta}{2^i}$, soit $n.\frac{1}{n^5} = \frac{1}{n^4}$.

□

Q 7. L'algorithme fonctionne en $\log \Delta + 1$ phases. Quelle est la probabilité pour qu'au moins une des phases se termine mal, c'est-à-dire qu'à l'issue d'au moins une des phases, au moins un nœud restant a un degré d'au moins $\frac{\Delta}{2^i}$ (si c'est la i^e phase) ?

Solution. même truc de preuve : on a "pour une phase", et on passe "à au moins durant l'une des phases"

La probabilité pour que ça se passe mal pour une des phases est au plus $\frac{1}{n^4}$. En appliquant l'inégalité de Boole, la probabilité pour que ça se passe mal pour au moins une des phases est bornée par $\frac{\log \Delta + 1}{n^4} \leq \frac{1}{n^3}$. □

Q 8. Conclure sur le fait que l'ensemble des sommets ajoutés à ce qu'on a appelé EIM est un ensemble indépendant maximal avec forte probabilité.

Solution. Par l'hypothèse de récurrence, à la fin de la phase $\log \Delta + 1$, chaque nœud a un degré d'au plus $\frac{\Delta}{2^{\log \Delta + 1}}$ avec forte probabilité. $\frac{\Delta}{2^{\log \Delta + 1}} \leq \frac{\Delta}{2.\Delta} = \frac{1}{2}$. Ceci implique que chaque nœud restant à la fin de l'algorithme a un degré de 0 (car le degré est un entier). Un nœud qui atteint un degré de 0 est ajouté à l'EIM. Et si un nœud a été enlevé avant la fin de l'algorithme, c'est parce que soit ce nœud a été ajouté à l'EIM, soit un de ses voisins a été ajouté à l'EIM. Donc l'ensemble des nœuds ajoutés à l'EIM est bien maximal avec forte probabilité. □