

Leçon 903 – Exemples d’algorithmes de tri. Correction et complexité.

8 juillet 2019

1 Extraits du Rapport

Rapport de jury 2018

Sur un thème aussi classique, le jury attend des candidats la plus grande précision et la plus grande rigueur. Ainsi, sur l'exemple du tri rapide, il est attendu du candidat qu'il sache décrire avec soin l'algorithme de partition et en prouver la correction en exhibant un invariant adapté. L'évaluation des complexités dans le cas le pire et en moyenne devra être menée avec rigueur : si on utilise le langage des probabilités, il importe que le candidat sache sur quel espace probabilisé il travaille. On attend également du candidat qu'il évoque la question du tri en place, des tris stables, des tris externes ainsi que la représentation en machine des collections triées.

Rapport de jury 2017

[idem] Le jury ne manquera pas de demander au candidat des applications non triviales du tri.

2 Cœur de la leçon

- Représentation des données : listes, tableaux.
- Propriétés des tris : stable, en place, en ligne.
- Algorithmes naïfs : insertion, sélection.
- Tri rapide, tri fusion.

3 À savoir

- Borne inférieure sur les tris par comparaison.
- Tri asymptotiquement optimaux : diviser-pour-régner (tri fusion), à base de structures de données (tri par tas).
- Analyses de complexité (meilleur, pire, moyenne, probabiliste) et preuves de correction.
- Tri linéaires (comptage, base).

4 Ouvertures possibles

- Tri rapide avec médian.
- Tri par ABR.
- Tim Sort.
- Réseaux de tri.
- Forcer un tri à être stable.
- Tris externe.

5 Conseils au candidat

- Il faut être rigoureux. Bien définir les opérations élémentaires autorisées.
- Pourquoi étudier les tris : intérêt pratique (utiles partout, et déjà codés dans des modules) mais surtout pédagogique (des algorithmes simples pour illustrer les notions de correction, de complexité, et les paradigmes algorithmiques).
- Illustrer la complexité par des exemples.
- Les algorithmes (qui sont simples) doivent être connus par cœur et sans hésitation.
- Analyses probabilistes du tri rapide : faire très attention aux probas. *Attention au Cormen*

6 Questions classiques

- Définir un « tri par comparaison » ? Pourquoi certains tris sont linéaires alors qu'on possède une borne inférieure ?
- Savez vous quels tris sont utilisés par votre langage de programmation favori ?
- Quelles sont les complexités des opérations élémentaires que vous prenez ?
- Quel tri utiliser en pratique en fonction de la taille des données ?
- Donner quelques applications non-triviales du tri ?
- Pourquoi veut-on un tri stable ? Un tri en place ? Un tri en ligne ?
- Comment améliorer le tri rapide pour qu'il soit optimal dans tous les cas ?

7 Références

- [Cor] Algorithmique - CORMEN - à la BU/LSV
La bible de l'algorithmique, avec toutes les bases. Attention, les calculs avec des probas sont parfois faux.
- [Bea] Éléments d'algorithmique - D. BEAUQUIER, J. BERSTEL, Ph. CHRÉTIENNE - à la BU/LSV
Bonne référence pour l'algo, pleins de dessins et de preuves. Un peu vieillissant et devenu rare.

8 Dev

- Complexité et correction du tri par tas - ([Cor], 3rd edition, p.154) - 901,903
Simple dans le fond, mais à bien faire formellement, et pédagogiquement. Dessins et exemples bienvenue.
- Complexité du tri rapide - ([Cor,Bea],) - 903,926,931
Bien faire attention au proba du Cormen, aller voir Beauquier est pertinent. Avoir une idée de l'écart type des performances.
- Borne inférieure sur la complexité d'un tri par comparaison - ([Cor],) - 903,926
Peut être un peu court.
- Tri bitonique - ([Cor],) - 903
Attention, disponible uniquement dans la seconde édition