Commentaires DS5 - CCINP 2024

Remarques générales :

- attention au respect des consignes : **pas de correcteur, marquer FIN à la fin de la copie**. C'est le cas pour les concours CCINP, Mines, Centrale (à cause de la numérisation des copies). X-ENS ?
- 1 ou 2 questions difficiles, mais peu de questions faciles. Il faut arriver à traiter des questions de difficulté "moyenne" : bien analyser les questions pour ne pas répondre "à côté"
- attention à la gestion du temps : il faut consacrer du temps à la 3ème partie
- expliquer ce que font les fonctions auxiliaires (qui étaient inutiles ici)

Exercice 1 - Algorithme Single Pass

Q1 - algorithme k-moyennes sur des triplets, avec une distance Manhattan. La moindre erreur rend le résultat faux.

Q2 - bien comprendre l'algorithme Single Pass : un document n'est classé qu'une fois : dans une classe qui lui est propre, ou bien rattaché à une classe existante (qui implique recalcul du centre) ... et encore une fois la moindre erreur rend le résultat faux. On trouve le même résultat qu'en Q1, mais en ayant fait beaucoup moins de calculs

Q3 - bien comprendre la structure de données qui est assez compliquée :

- vecteur (struct) : représente un document ou un centre
- double est le type réel sur 64 bits;
- v désigne un tableau alloué dans le tas, de longueur taille
- num_classe : pour associer une classe à un document
- noeud (struct) : noeud d'une liste chaînée qui sert à représenter les centres

ajoutVecteur : le paramètre tete sert à la fois de paramètre d'entrée ET de paramètre de sortie : *tete désigne la liste avant ajout ; et après ajout, on met à jour le pointeur pointé par tete : *tete = ...

NB : vérifier que l'allocation mémoire s'est bien passée = vérifier que le pointeur renvoyé par malloc est non NULL

Q4 - très facile : en C la fonction abs n'est pas disponible directement : à implémenter, ou gérer le signe "à la main"

Q5 - très facile : penser à utiliser return dans la boucle forcément

Q6 - très facile : penser à prendre comme valeur de référence dists[0]

Q7 - assez difficile : avec plusieurs points délicats :

- les centres sont gérés par une liste chaînée,
- il faut gérer correctement le décalage des indices : l'algo numérote les documents à partir de 1, le programme à partir de 0
- le tableau des j distances doit être alloué dans le tas (j n'est pas une constante) et donc libéré
- les centres sont stockés "à l'envers" dans la liste chaînée.

Ne pas pas oublier la complexité, à justifier

Exercice 2 - Langages réguliers

Bien comprendre les définitions (s'aider de l'exemple)

- R : deux mots sont en relation s'ils sont égaux à un échange $ab \leftrightarrow ba$ près
- R^* : deux mots sont en relation s'ils sont égaux à plusieurs échanges près

Q8 - relation d'équivalence : réflexivité, symétrie, transitivité. On raisonne avec la définition de R^* (inutile de revenir à la définition de R). Attention, il y a 2 cas ($u=v$ ou il existe des mots ...)

Q9 - astuce : ici, une classe d'équivalence est un ensemble fini, donc régulier !

Q10 - les classes d'équivalence sont des ensembles de mots qui forment une partition de l'ensemble des 27 mots possibles. (on trouve 21 classes)

Q11 - on reconnaît l'ordre lexicographique. transitivité : il y a 4 cas à gérer. Pour justifier "ordre" : il faut justifier l'antisymétrie. Pour justifier "total" : il faut justifier que deux mots quelconques sont comparables

Q12 - assez facile

Q13-Q14 - il faut un peu d'astuce : les représentants s'écrivent d'une manière précise (pas de a après un b) et on en déduit l'automate

Problème - Correspondance de Burge

Beaucoup de définitions que l'on s'approprie au fur et à mesure du sujet.

d_{d+1} à bien comprendre : puisqu'il y a un sommet de degré d , il y a au moins $d+1$ sommets (ce sommet et ses voisins) donc d_{d+1} est bien défini.

Q15 - facile : lemme des poignées de main

Q16 - Q17 - on demande de démontrer le théorème. Q16 assez facile, il suffit de rajouter un sommet de degré d et de le relier aux d sommets de plus grand degré. Q17 difficile

Q18 - il suffit de justifier que la suite est décroissante

Q19 - on utilise le théorème, en supprimant le sommet de degré maximal

Q20 - très facile

Q21 - Q22 - plus difficile, mais se fait bien par récurrence (OCaml impératif interdit par le sujet). type option à savoir manipuler (par un match !)

Q23 - assez facile, on demande 2 graphes différents !

Q24 à Q26 - Encore des définitions ... mais c'est assez facile

Q27 - on demande juste un invariant pertinent (pas de prouver que l'algorithme est correct)

Q28 - Q29 - Q30 - programmation OCaml : pas facile, mais tout à fait faisable (on procède par récurrence)

Q31 à Q35 - assez facile, mais il faut bien s'être approprié tout ce qui précède