ESIR1 / ALGC mardi 15 avril 2014
Contrôle continu 30 mn

Répondre de façon concise mais précise aux questions suivantes en utilisant les cadres. Aucun document n'est autorisé, hormis un aide-mémoire manuscrit de format A4 recto seul constitué par chaque étudiant. Ce sujet comporte 2 pages. Ne pas oublier d'indiquer ses noms et prénoms.

Éléments de réponses

Quelle est la différence entre un problème et un algorithme ?
Qu'est ce que ne pas être décidable ?

Considérez d'abord qu'il s'agit d'une question de cours, avec document, sur le titre même du module. Ne pas savoir y répondre est scandaleux. Un ingénieur informaticien ne peut pas n'être qu'un programmeur, il doit être aussi quelqu'un qui connaît les limites de la programmation, le pourquoi de ces limites, et la façon de s'en approcher le plus possible.

Un problème est l'énoncé d'une fonction par sa relation entre entrées et sorties, en extension donc. Un algorithme est l'énoncé d'une fonction par l'expression d'une méthode de calcul. Toute l'algorithmique tourne autour de ça. Est-il toujours possible de trouver un algorithme pour résoudre un problème, ou est-ce qu'il y a des limites à ce qu'on peut calculer ? Est-il toujours possible de trouver des algorithmes plus efficaces, ou est-ce qu'il y a des limites infranchissables ? etc...

Quelle est la remarque fondamentale qui entraine que tous les problèmes ne sont pas décidables ? Mentionner les théorèmes qui fondent cette remarque.

La remarque est qu'il y a beaucoup plus de problèmes que d'algorithmes. Elle dérive de deux théorèmes de Georg Cantor : la dénombrabilité de l'ensemble des rationnels, et l'indénombrabilité de l'ensemble des fonctions sur les entiers.

Citer trois problèmes indécidables, dont le premier à avoir été prouvé tel.

Le problème de l'Arrêt (le premier), toutes ses variantes, l'équivalence de programmes, le test à zéro, le problème de Post, Entscheidungsproblem, tous les problèmes concernés par le théorème de Rice, …

Soit deux problèmes A et B, A est connu pour être indécidable. Comment prouver l'indécidabilité de B en s'appuyant sur celle de A ?

Répondre « en faisant une réduction » n'avance pas à grand chose puisque tout l'enjeu est de savoir dans quel sens la faire. Il faut la faire de A vers B (A ≤c B) car les mauvaises nouvelles vont de gauche à droite… Si on peut réduire P en Q, alors Q est au moins aussi difficile que P puisqu'une solution de Q est aussi une solution de P. Donc si on peut réduire calculatoirement A en B, on montre que B est au moins aussi difficile que A, donc non-décidable.

En quoi consiste l'étude de la complexité asymptotique ?

C'est l'étude du coût de l'exécution d'un algorithme quand on pousse à la limite une dimension des données qu'il traite. Asymptote = vers la limite, comme en analyse. Dans ces conditions on peut s'affranchir de nombreux détails, comme les facteurs constants ou les composants d'ordre inférieur du coût.

Supposons que $f = O(g)$. Rayer celles des affirmations suivantes qui sont fausses.

1. **$f + g = O(g)$.** Vrai puisque g domine f.
2. **si $g = O(h)$ alors $f = O(h)$.** Vrai transitivement.
3. **Le rapport $f(x) / g(x)$ tend vers 1 quand x croît.** Faux, dominer n'est pas converger vers.
4. **Le rapport $f(x) / g(x)$ est borné quand x est suffisamment grand.** Vrai, par définition.
5. **$O(f)$ est inclus dans $O(g)$.** Vrai, par définition.
6. **$O(f+g)$ est inclus dans $O(g)$.** Vrai, ils sont même égaux.
7. **$g = \Theta(f)$.** Faux, ça voudrait dire que f domine g aussi et on n'a rien dit de tel.
8. **f est une fonction polynomiale.** Faux, ça n'a rien à voir.

Rappeler la différence entre problème de décision et problème d'optimisation et le rôle que chacun d'eux joue par rapport à l'étude des algorithmes.

Là encore j'ai eu trop de réponse de programmeurs et pas assez d'ingénieur. Ce cours n'a jamais parlé d'optimisation de programme ! Un problème de décision retourne une valeur dans {vrai, faux}. Un problème d'optimisation retourne une valeur qui est la meilleure selon un critère donné dans l'énoncé du problème. Ex. trouver si il existe un chemin de longueur inférieure à K est un problème de décision, trouver le plus petit chemin est un problème d'optimisation.

Quelle est la formule de récurrence de la complexité de l'algorithme de Karatsuba ? Quelle est la complexité de cet algorithme ?

Voir le cours. Karatsuba = multiplication rapide, dont le coût est dans O(n**log2 3) = O(n**1,59), C'est le premier exemple de fast-truc, obtenu par diviser pour régner, et qui donne ces complexités bizarre avec exposants irrationnels. Il est important de savoir que de telles complexités existent et pourquoi.

Soit F et G deux problèmes tels que $coût(F) = O(x^k)$ et $coût(G) = \Theta(a^x)$. Soit I le problème qui à chaque entrée associe une sortie égale à l'entrée. Rayer celles des affirmations suivantes qui sont fausses.

1. **Il est possible que F se réduise calculatoirement en G.**
2. **Il est impossible que G se réduise calculatoirement en F.**

La réduction calculatoire ne parle pas de coût, donc tout est possible ; 1 est vrai, et 2 et faux. Attention, possible ne veut pas dire certain, seulement qu'il y a des cas où c'est vrai. Exemple, G=calculer la table de vérité d'une formule propositionnelle est dans $\Theta(2^x)$; d'après le théorème de Cook tout problème de NP, donc de P, donc F, peut se réduire polynomialement en G. Inversement, G peut se réduire calculatoirement en beaucoup de chose dont I, de la façon suivante : G = I o I o G . Pour la réduction calculatoire, tout est permis pour les fonction f et g, du moment qu'elles soient calculables.

3. **Il est possible que F se réduise polynomialement en I.**
4. **Il est impossible que I se réduise polynomialement en G.**

3 est vrai, car F = I o I O F, et I est polynomiale. 4 est faux, car il suffit de faire faire quelque chose d'inutile à G ; f = i -> (i-ème formule propositionnelle, i), G = (F, i) -> (F est satisfaisable, i), g = (b, i) -> i.

Conclusion : ce contrôle continu a comporté une majorité de questions de cours (avec document) qui ne demandaient aucune réflexion, et pourtant la moyenne a été de 8 ! Rentrez plus dans ce cours, bossez les définitions, bossez les raisonnements (ils sont souvent subtiles, mais c'est comme ça), faites ça à plusieurs, recherchez des algorithmes hors du cours d'algorithmique. Réfléchissez en faisant votre aide-mémoire. Celui-ci ne dispense pas d'apprendre, il vous aide à apprendre, encore faut-il s'y prendre sérieusement.