Master Informatique M1

2018/2019

Algorithme et complexité
Partie 2

Thiebaud MARBACH
Alexis ABLI-BOUYO

Répartition des tâches:

Bien que la plupart des réflexions de la Partie 1 ont été faites en binômes, le développement a été réparti de la façon suivante:

***Thiebaud**:* Algo de Kruskal et Algo d'Aldous-Broder
Alexis: Algo de Wilson et Gestion des labyrinthes

La partie 2 a été totalement faite en binôme.

Q9) Il y a un total de A_K^N réponses possibles au maximum avec A_X^Y le nombre d'arrangements à Y éléments parmi X.

Q10) (b,0) signifie qu'il y a b couleurs bien placées et aucune au mauvais endroit. Nous avons donc b parmis N possibilités pour placer les bonnes réponses dans notre solution proposée, soit C_N^b avec C_X^Y le nombre de parties à Y éléments parmi X.
Pour chaque cas de figure, il y a plusieurs façon d'essayer les couleurs restantes (en sachant qu'il y a K-N jamais utilisées) dans chacunes des possibilités (en sachant qu'on a N-b emplacements où nous devons tester de nouvelles couleur).
Il y a donc un total de A_{K-N}^{N-b} possibilités pour les emplacements restants.
Il y a donc au maximum C_N^b * A_{K-N}^{N-b} possibilités secrètes possibles en prenant en compte une réponse (b,0).

Q11) Nous avons b parmis N possibilités pour placer les b bonnes réponses dans notre solution proposée, soit C_N^b.
Nous avons m parmis N-b emplacements pour positionner les m réponses initialement mal placées en partant du principe que nos b couleurs sont déjà bien placées, soit C_{N-b}^m.
Pour chaque cas de figure, il y a plusieurs façon d'essayer les m couleurs initialement mal placées dans chacunes des possibilités (en sachant qu'on a N-b emplacements où nous devons tester de nouvelles couleurs).
Il y a donc un total de A_{N-b}^m possibilités pour les emplacements restants.
Pour chaque cas de figure, il y a plusieurs façon d'essayer les couleurs restantes (en sachant qu'il y a K-N couleurs jamais utilisées) dans chacunes des possibilités (en sachant qu'on a N-b-m emplacements où nous devons tester de nouvelles couleurs).
Il y a donc un total de A_{K-N}^{N-b-m} possibilités pour les emplacements restants.

Il y a donc au maximum C_N^b * C_{N-b}^m * A_{N-b}^m * A_{K-N}^{N-b-m} possibilités secrètes possibles en prenant en compte une réponse (b,m).

Q12) Il n'y a pas d'égalité car, avec ce raisonnement, on peut obtenir de plusieurs façons une possibilité.

Q13) C(k, n, b, m)
Formules en fonction de la première couleur proposée:
bien placée C(k-1,n-1,b-1,m)
mal placée C(k-1,n-1,b,m-1)
absente C(k-1,n,b,m)

avec le nombre de combinaisons secrètes de taille n parmi k couleurs qui sont compatibles avec la réponse (b, m) fournie à une proposition de taille n

Nous n’avons pas réussi à obtenir une formule de récurrence.

Q14 et Q15) N’ayant pas réussi à obtenir une formule de récurrence à la question 13, nous avons créé un algorithme récursif, probablement plus complexe.

Pour chaque étape du parcours récursif, on regarde une case de la proposition. On distingue les dix cas suivants:

Si la couleur de la case est à sa place:
- 1) La couleur de la case est **bien placée**.

Si la couleur de la case est mal placée:
- 2) La couleur de la case a sa place à **droite**; on prend une couleur **absente** de la proposition pour la mettre dans cette case
- 3) La couleur de la case a sa place à **droite**; on prend une couleur qui était mal placée à **droite** de cette case pour la mettre dans cette case
- 4) La couleur de la case a sa place à **droite**; on prend une couleur qui était mal placée à **gauche** de cette case pour la mettre dans cette case

- 5) La couleur de la case a sa place à **gauche**; on prend une couleur **absente** de la proposition pour la mettre dans cette case
- 6) La couleur de la case a sa place à **gauche**; on prend une couleur qui était mal placée à **droite** de cette case pour la mettre dans cette case
- 7) La couleur de la case a sa place à **gauche**; on prend une couleur qui était mal placée à **gauche** de cette case pour la mettre dans cette case

Si la couleur de la case n’apparait pas dans la solution:
- 8) La couleur de la case est **absente** dans la solution; on prend une couleur **absente** de la proposition pour la mettre dans cette case
- 9) La couleur de la case est **absente** dans la solution; on prend une couleur qui était mal placée à **droite** de la case pour la mettre dans cette case
- 10) La couleur de la case est **absente** dans la solution; on prend une couleur qui était mal placée à **gauche** de la case pour la mettre dans cette case

On part du nombre de cases non traitées N, le nombre de couleurs encore inutilisées K, le nombre de bonnes couleurs B, de mauvaises couleurs M et de couleurs absentes A; on compte également le nombre de mauvaises couleurs qu’on essaie de déplacer vers la droite MD et le nombre de mauvaises couleurs qu’on essaie de déplacer vers la gauche MG.
Pour chaque étape, on regarde une case, on regarde les cas qui peuvent être appliqués selon les valeurs de B, M et A. Pour chaque cas qui est possible, on va faire un ou plusieurs appels récursifs, en augmentant ou diminuant les valeurs en paramètre.

Par exemple pour le cas 9, on dit que la couleur de la case est absente de la solution et n'interviendra plus. On attend donc une couleur mal placée venant de la droite (une couleur qui sera envoyée vers sa gauche, donc comptée par mauvaisesGauche MG). L'appel est donc une fois parcours(couleursInutilisees, casesRestantes-1, bonnes, mauvaises, absentes-1, mauvaisesGauche+1, mauvaisesDroite).
Pour le cas 6, on appelle l'appel récursif pour chaque fois qu'on a trouvé une position qui pouvait attendre une couleur mal placée venant de sa droite, et pour chaque fois qu'on a trouvé une couleur mal placée pouvant être déplacée vers la droite. L'appel récursif dans ce cas est parcours(couleursInutilisees, casesRestantes-1, bonnes, mauvaises-1, absentes, mauvaisesGauche-1, mauvaisesDroite -1).

La condition d'arrêt est: si les valeurs de B, M, A, MD et MG valent toutes zéro lors d'une étape, c'est que la combinaison est correcte. On incrémente donc le compteur de combinaisons possibles.

Puisque chaque couleur ne peut apparaître que 0 ou 1 fois dans la solution et dans la proposition, la couleur elle-même n'importe pas dans le raisonnement tant qu'on trouve sa position finale ou qu'on l'enlève de la solution. De ce fait, l'algorithme ne nécessite pas de stocker les positions des différentes couleurs.

L'algorithme est implémenté dans la classe Partie2Q15 du projet. On peut l'éxecuter grâce à la méthode main de cette classe.
L'explication peut sembler confuse mais l'implémentation est plus claire.
En utilisant notre implémentation, on calcule que C(6,4,1,2) = 72.

```
K?
6
N?
4
B?
1
M?
2
C(6,4,1,2)=72
```

Q16) La complexité théorique de notre algorithme au pire cas devrait valoir O(n) = $10^{(n-1)}$. La complexité moyenne devrait valoir bien moins que ca vu que de nombreux cas sont incompatibles entre eux, mais devrait rester exponentielle.

Q17) Il est difficile d'effectuer les calculs des questions précédentes avec un historique non vide car les informations de l'historique rendent impossible un certain nombre de cas à l'étape actuelle.

Q18) H est la quantité de reponse possible en tenant compte de toute les propositions de l'historique et b,p,m le nombre de réponses possibles en tenant seulement compte de notre dernière proposition.
L'objectif est donc de minimiser le score en proposant les meilleures réponses possibles.
Nous n'avons pas eu le temps de faire marcher la question 18 vu que nous avons eu beaucoup de mal à obtenir une réponse correcte pour la question 15.