

M1.L2: Série d'exercices sur les algorithmes [solutions]

1 Algorithmes de conversion

1.1 Conversion virgule fixe vers décimal

On obtient l'accumulation dans X des valeurs suivantes $0.25 + 0.5 + 2 + 8 = 10.75$

Le nombre total d'opération ne dépend pas de n ni de la valeur des éléments de P. Dans tous les cas il y a 32 passages dans l'itération car i varie de 1 à 32.

1.2 Conversion entier décimal vers binaire

Valeur de n	motif binaire obtenu	nombre de passages
0	0	1
1	1	1
2	10	2
5	101	3
10	1010	4
33	100001	6
80	1010000	7

Le `nb_de_passage` est proche de la puissance de 2 supérieure ou égale à n , qu'on obtient avec $\text{partieEntière}(\log_2(n)) + 1$

1.3 Conversion décimal fractionnaire vers binaire

Valeur de a	motif binaire obtenu
0.5	0.1
0.25	0.01
0.375	0.011

0.2 est en fait deux fois un dixième. Or un dixième a une longueur infinie en binaire. Il faut donc s'arrêter à 32 bits. On peut aussi reprendre le motif binaire de un dixième et le décaler d'un cran vers la gauche :

0.00110011001100110011001100110011001100

Contrairement à la question 1.2, il n'est pas trivial d'estimer le nombre de bits que va occuper la représentation binaire d'un nombre compris entre 0 et 1.

1.4 Conversion virgule flottante du cours M1.L1 vers décimal

- $F(4) = 1$, $F(3) = 0$, $F(2) = 0$, $F(1) = 1$: l'exposant vaut 2, la mantisse 0.25. il s'agit de la forme normalisée, la quantité représentée est donc : $2^2 * 1.25 = 5$
- $F(4) = 0$, $F(3) = 0$, $F(2) = 0$, $F(1) = 1$: l'exposant vaut 0, la mantisse 0.25. il s'agit de la forme dénormalisée, la quantité représentée est donc : $2^1 * 0.25 = 0.5$
- $F(4) = 0$, $F(3) = 0$, $F(2) = 0$, $F(1) = 0$: c'est la représentation de zéro

1.5 Conversion d'un nombre décimal vers virgule flottante

- 9.5 : l'exposant vaut 3 c'est-à-dire 11 en binaire
la division de 9.5 par 8 donne en binaire 1.0011 , or seuls les deux premiers bits peuvent être mémorisés dans la mantisse qui vaut donc 00. Il s'agit d'une approximation. Sa représentation est 8, l'erreur absolue vaut 1.5 et l'erreur relative vaut $1.5/9$ c'est-à-dire environ 16%.
- 13 : l'exposant vaut 3 c'est-à-dire 11 en binaire
la division de 13 par 8 donne en binaire 1.101 , or seuls les deux premiers bits peuvent être mémorisés dans la mantisse qui vaut donc 10. Il s'agit d'une approximation. Sa représentation est 12, l'erreur absolue vaut 1 et l'erreur relative vaut $1/13$ c'est-à-dire environ 7.7%.
- 1.25 : cette valeur est inférieure à 2, il doit utiliser la forme **dénormalisée** pour laquelle l'exposant vaut 0 c'est-à-dire 00 en binaire. On utilise $p=1$ pour le calcul de la mantisse.
la division de 1.25 par 2 donne en binaire 0.101 , or seuls les deux premiers bits peuvent être mémorisés dans la mantisse qui vaut donc 10. Il s'agit d'une approximation. Sa représentation est 1, l'erreur absolue vaut 0.25 et l'erreur relative vaut $0.25/1.25$ c'est-à-dire 20%.
- 5 : l'exposant vaut 2 c'est-à-dire 10 en binaire
la division de 5 par 4 donne en binaire 1.01 , la mantisse qui donc 01. Ce nombre est exactement représenté ; l'erreur absolue et l'erreur relative sont nulles.

La précision de cette représentation est donnée par le poids faible de la mantisse c'est-à-dire 0.25 ou 25%. C'est un majorant des erreurs relatives qui peuvent exister pour les nombres représentés avec la **forme normalisée**.

Malheureusement, l'erreur relative peut augmenter au-delà de 25% dans le petit intervalle traité avec la **forme dénormalisée** mais c'est le prix à payer pour inclure zéro dans cette représentation.

2 Calculs mystérieux 1

Devinette1 :

Exemple de déroulement pour $a = 5$ et $b = 7$:

a	b	c
5	7	0
10	3	5
20	1	15
40	0	35

Exemple de déroulement pour $a = 10$ et $b = 9$:

a	b	c
10	9	0
20	4	10
40	2	10
80	1	10
160	0	90

Cet algorithme permet de calculer le produit entre deux nombres entiers. C'est la forme algorithmique de la multiplication égyptienne abordée « manuellement » la semaine passée.

https://fr.wikipedia.org/wiki/Technique_de_la_multiplication_dans_l%27%C3%89gypte_antique

3 Calculs mystérieux 2

Exemple de déroulement pour $x = 10$ et $y = 25$:

x	y
35	25
10	25
10	15
10	5
5	5

Exemple de déroulement pour $x = 70$ et $y = 42$:

x	y
70	42
28	42
28	14
14	14

c'est l'algorithme d'Euclide calculant le PGCD (*plus grand commun diviseur*) des deux entiers x et y

https://fr.wikipedia.org/wiki/Algorithme_d%27Euclide

4 Création d'algorithmes

4.1 La plus petite valeur

Il suffit de regarder tour à tour toutes les valeurs, en mémorisant la plus petite vue jusqu'ici :

La plus petite valeur
entrée : Liste L d'entiers non vide, de taille n sortie : a , plus petite valeur dans L
$a \leftarrow L(1)$ (premier élément de L) Pour i de 2 à n Si $a > L(i)$ $a \leftarrow L(i)$ sortir : a

Cet algorithme effectue $n-1$ comparaisons. Il fonctionne aussi bien avec des valeurs ex aequo.

4.2 La plus petite différence

Une approche possible consiste à tester les différences entre toutes les paires possibles :

La plus petite différence-1
entrée : Liste L d'entiers, de taille n , contenant au moins 2 éléments sortie : a, b nombres avec la plus petite différence
$a \leftarrow L(1)$ $b \leftarrow L(2)$ $\delta_{min} \leftarrow L(1) - L(2) $ Pour i de 1 à $n-1$ Pour j de $i+1$ à n $\delta \leftarrow L(i) - L(j) $ (valeur absolue) Si $\delta < \delta_{min}$ $\delta_{min} \leftarrow \delta$ $a \leftarrow L(i)$ $b \leftarrow L(j)$ sortir : (a, b)

Estimons le nombre de comparaisons effectuées par cet algorithme. Soit n le nombre d'éléments dans la liste :

il y a 2 boucles imbriquées dans l'algorithme : le nombre total m d'itérations effectuées par la boucle intérieure (donc le nombre de comparaisons) est donné par :

$$m = 1 + 2 + 3 + \dots + (n - 1) = n(n - 1)/2 = \frac{1}{2}(n^2 - n)$$

On voit dans cette expression un terme en n^2 ; c'est le point ce qui distingue cet algorithme du précédent qui était simplement proportionnel à n .

5 PageRank (optionnel)

Après l'insertion d'un seul lien vers le blog de Rosa, son score PageRank, noté S_R , sera donné par l'équation :

$$S_R = 0.15 + 0.85 \frac{S}{L}$$

où S est le score PageRank de la page pointant vers le blog de Rosa, et L est le nombre de lien sortant de cette page. Rosa doit donc choisir la page qui a le plus grand quotient S/L . Pour déterminer cela, il nous faut calculer le PageRank de chaque page web.

Commençons par écrire leurs équations. Soient S_G le score de George.com, S_F le score de Fabienne.fr, S_S le score de Sofien.ch et S_W le score de Wolfgang.me, et soit L_G le nombre de liens sortants de George.com, L_F le nombre de liens sortants de Fabienne.fr, etc. Nous avons alors :

$$L_G = L_S = 1 \qquad L_W = L_F = 2$$

et :

$$S_W = 0.15$$

$$S_G = 0.15 + 0.85 \times (S_W/2 + S_F/2 + S_S)$$

$$S_F = 0.15 + 0.85 \times S_G$$

$$S_S = 0.15 + 0.85 \times (S_W/2 + S_F/2)$$

Si l'on applique la méthode itérative de PageRank, on obtient :

	étape 1	étape 2	étape 3	étape 4	étape 5	...
S_W	0.25	0.15	0.15	0.15	0.15	...
S_G	0.25	0.57500	0.67594	0.79786	0.93412	...
S_F	0.25	0.36250	0.63875	0.72455	0.82818	...
S_S	0.25	0.36250	0.36781	0.48522	0.52168	...

Vous constaterez que l'on met du temps à converger vers la solution, ce qui rend le calcul assez rébarbatif...

Nous vous avons demandé le calcul au bout de trois itérations. On a à ce stade :

$$S_S/L_S = 0.36781, \quad S_F/L_F = 0.31938, \quad S_W/L_W = 0.075, \quad S_G/L_G = 0.67594$$

Pour maximiser son score PageRank, Rosa devrait donc choisir de demander à George d'insérer un lien de George.com a son blog.

Si l'on avait fait le calcul exact (non demandé), nous aurions eu :

$S_S/L_S = 0.83663$, $S_F/L_F = 0.77389$, $S_W/L_W = 0.075$, $S_G/L_G = 1.54777$ et serions arrivés à la même conclusion.

Aparté

Si l'on veut être sûr, on pourrait ici résoudre les équations. Ce n'est pas ce que nous vous demandons ! mais nous vous l'indiquons pour information :

$$S_W = 0.15$$

$$S_G = 0.15 + 0.85 \times (S_W/2 + S_F/2 + S_S) = 0.21375 + 0.425S_F + 0.85S_S$$

$$S_F = 0.15 + 0.85 \times S_G$$

$$S_S = 0.15 + 0.85 \times (S_W/2 + S_F/2) = 0.21375 + 0.425S_F$$

et donc (soustraction de 2^e et 4^e équations)

$$S_G = S_S + 0.85S_S = 1.85S_S$$

donc

$$S_F = 0.15 + 1.5725S_S$$

qui nous conduit a

$$S_S = 0.21375 + 0.063750 + 0.66831S_S$$

d'où

$$S_S = 0.83663 \quad S_G = 1.54777 \quad S_F = 1.46560$$