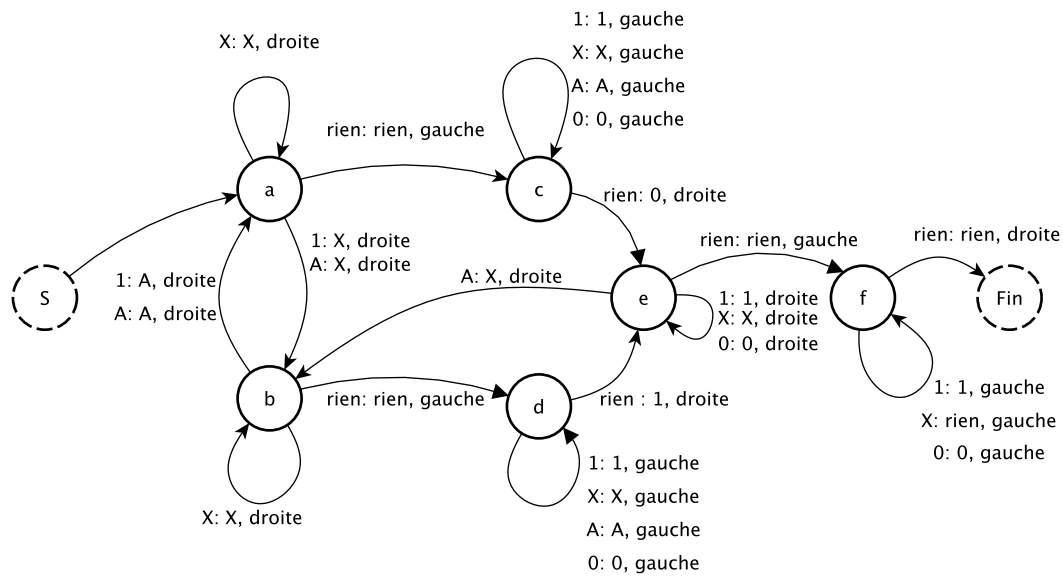


1. La question suivante touche les tableaux de hachage
  - a. En quoi le choix de la fonction de hachage a un impact sur le nombre de collisions? Donner un exemple d'une fonction de hachage qui en produirait beaucoup
  - b. Dans le cas où les collisions sont gérées par des listes, est-ce qu'il est absolument nécessaire d'augmenter la taille du tableau lorsqu'il est trop plein? Considérez d'abord que vous désirez minimiser l'espace mémoire requis. Si l'objectif est de permettre de déterminer le plus rapidement possible si une clé est déjà utilisée, est-ce que cela change votre réponse? Expliquer.
2. . Soit un arbre binaire AVL, dans lequel on insérera les entiers de 1 à 7
  - a. Quelle est la hauteur de l'arbre idéal (parfaitement balancé)?
  - b. En supposant que vous pouvez imposer l'ordre des nombres à insérer, quel ordre choisiriez-vous pour minimiser le nombre de rotations?
  - c. Quel ordre pourriez-vous utiliser pour maximiser le nombre de rotations, et quel serait alors le nombre de rotations effectuées (à la fin de la création de l'arbre)?

3. Soit la machine de Turing suivante. Déterminer quel calcul elle effectue. Le ruban est de longueur infinie, les trois symboles suivants sont utilisés : {0, 1, rien}. On suppose que le ruban démarre avec la séquence : (1, 1, 1), avec la tête de lecture positionnée sur le « 1 » le plus à gauche. Refaire le calcul avec (1,1,1,1,1) comme séquence de départ.



Étape	État	Entrée (valeur ruban)	Écriture	Mouvement	Prochain état	Ruban - machine de Turing														
1	S	1	1	-	a					1	1	1								
2	a	1	X	droite	b					1	1	1								
3	b	1	A	droite	a					x	1	1								
4	a	1	X	droite	b					x	a	1								
5	b	r	r	g	d					x	a	x								
6	d	x	x	g	d					x	a	x								
7	d	a	a	g	d					x	a	x								
8	d	x	x	g	d					x	a	x								
9	d	r	1	d	e					x	a	x								
10	e	x	x	d	e					1	x	a	x							
11	e	a	x	d	b					1	x	a	x							
12	b	x	x	d	b					1	x	x	x							
13	b	r	r	g	d					1	x	x	x							





4. La question qui suit touche les algorithmes de tri.
  - a. Quelle est la complexité du tri à bulle? Expliquer votre réponse
  - b. Dans le tri rapide (quicksort), expliquer pourquoi le choix du pivot est important.
  - c. Pour un ensemble de N nombres, indiquer quelle quantité de mémoire sera nécessaire pour effectuer un tri fusion.
  
5. Quelle est la complexité 'Big O' des extraits de code suivants?
  - a. 

```
for i in xrange(1, 10000) :
    for j in xrange (1, 20000) :
        dict[i+j]= i+ j
```

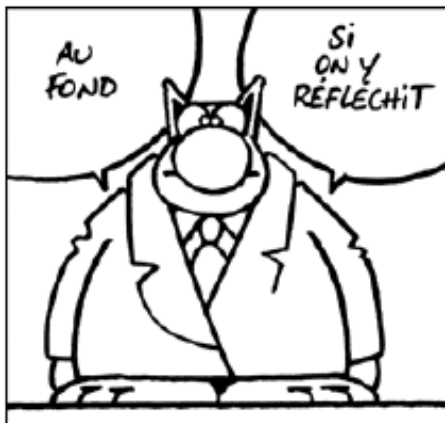
1
  
  - b. 

```
def fonction1 (a) :
    if a <= 1:
        return
    fonction1 (a - 10)
```

n
  
  - c. 

```
def fonction2 (a) :
    if a <= 1:
        return
    fonction2 (a / 10 + 5)
```

logn
  
6. Soit un graphe G avec V nœuds et E arcs ,  $G(V,E)$ . Dans le meilleur des cas, combien de nœuds doivent être visités pour réaliser l'algorithme de Dijkstra?
  
7. Cette question touche des considérations générales au sujet des structures de données
  - a. Supposons qu'on désire emmagasiner une quantité N de données. Est-il possible de créer la structure contenant l'ensemble des données en  $O(\log N)$  opérations? Pourquoi?
  - b. Si les données, une fois entrées dans le système, seront toujours traitées en entier (lors d'une passe de traitement, toutes les données seront toujours touchées), quelle structure de données proposeriez-vous, et pourquoi?
  - c. Si de très grandes quantités de données seront emmagasinées (des centaines de milliards de données), et qu'il faudra retrouver une donnée particulière lors des traitements subséquents, quelle structure de données proposeriez-vous, et pourquoi?



(Tiré de « Le chat » de Geluck)