mystyle backgroundcolor= commentstyle=, keywordstyle=, numberstyle=,
stringstyle=, basicstyle=, breakatwhitespace=false, breaklines=true, captionpos=b, keepspaces=true,
numbers=left, numbersep=5pt, showspaces=false, showstringspaces=false, showtabs=false,
tabsize=2

style=mystyle

Midterm Algo 2027

1 Questions de cours

1. Donnez une méthode de hachage direct: chainage séparé ou hachage coalescent
2. Le problème qui apparait avec le hachage coalescent: des collisions secondaires peuvent apparaitre
3. Une collision primaire est: une coïncidence de valeur entre h(x) et h(y) tel que x≠y, avec h la fonction de hachage
4. L'ordre d'un graphe non-orienté est: le nombre de sommet du graphe
5. Un sommet isolé est: un sommet de degré num
6. Le tableau des demi-degrés extérieurs des sommets de G:

1	2	3	4	5	6	7	8	9
2	2	3	2	0	2	3	2	2

7. Le graphe est-il fortement connexe: Non
8. Si non, combien possède t-il de composantes fortement connexes? 2
9. S'ils existent, les sommets de G de degré différent de 4 sont: 1,5 et 9
10. S'ils existent, les sommets de G de demi-degré intérieur égal à 0 sont: ∅

2 Arbres généraux

[language=Python] def keys$_a ft er_b is(T, x)$: $q = queue.Queue()q.enqueue(T)q_next = queue.Queue()found = FalseL = []whilenotq.isempty()$: $whilenotq.isempty()andnotfound$: $T = q.dequeue()ifT.key == x$: $found = Trueelse$: $forCinT.children$: $q_next.enqueue(C)iffound$: $whilenotq.isempty()$: $L.append(q.dequeue().key)else$: $(q, q_next) = (q_next, q)returnL$

[language=Python] def $_test_subtrees(B) : (nb, key_sum) = (1, B.key)C = B.childwhileC : (ok, n, s) =_t est_subtrees(C)ifnotokors/n > B.key : return(False, nb, key_sum)nb+ = nkey_sum+ = sreturn(True, nb, key_sum)$

def average$_subtrees(B) : return_test_subtrees(B)[0]$

3 B-arbres