Université Hadj Lakhdar -Batna-
Faculté des sciences
Département d'informatique

Batna le 31/05/2014

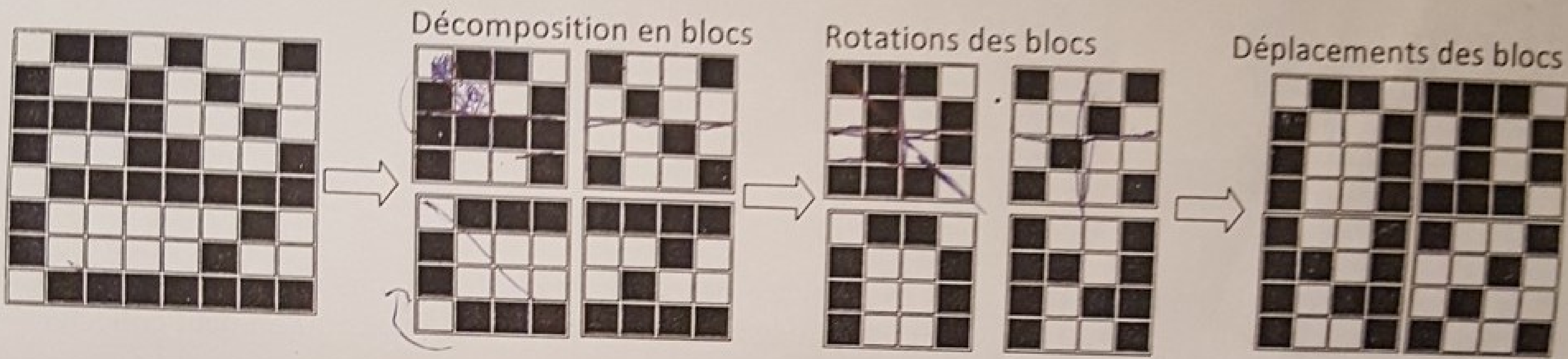
Matière : Algorithmique avancée
et preuve des programmes

Contrôle final (1ère année Master IRC)

Durée 2 heures

Exercice 1 (10pts)

Une manière de faire pivoter de 90° une forme représentée par une matrice de $n \times n$ pixels ($n = 2^k$) est de diviser la matrice en quatre blocs de $n/2 \times n/2$ pixels chacun, d'effectuer une rotation de 90° à chacun de ces blocs, puis de les déplacer à leurs places appropriées comme indiqué ci-dessous :



a) A l'aide de quel paradigme ce problème est-t-il résolu ? Expliquer.
b) Montrer comment faire pivoter de 90° un des quatre sous blocs par applications récursives des étapes précédentes (choisir un sous bloc quelconque).
c) Sur la base de ce qui précède, écrire un algorithme récursif qui permette d'effectuer une rotation de 90° à une forme représentée par une matrice de $n \times n$ pixels ($n = 2^k$). Pour déplacer un bloc de pixels on peut utiliser une fonction de la forme Memcopy(&Source, &Destination, Longeur).
d) Calculer la complexité de cet algorithme dans le pire des cas.

Exercice 2 (5pts)

Deux algorithmes A et B mettent exactement $T_A(n) = 0.1n^2 log_{10}n$ et $T_B(n) = 2.5n^2$ microsecondes, respectivement, pour traiter un problème de taille n .

a) Choisir le meilleur algorithme au sens Big-O.
b) Trouver une taille de problème n_0 telle que pour toute taille $n > n_0$,l'algorithme choisi dépasse en performance l'autre algorithme.
c) Si nos problèmes sont tous de tailles $n \leq 10^9$, quel algorithme allez vous recommander.

Exercice 3 (5pts)

Utiliser l'algorithme de parcours en profondeur d'abord (DFS) pour trouver tous les cycles d'un graphe connecté, s'il en existe.

Exercice 4 (5pts)

On vous donne une procédure $Sort(k, k + 10)$ qui trie les dix (10) éléments $A[k], A[k + 1], \ldots, A[k + 10]$, d'un tableau $A[1 \ldots n]$ donné ($n \geq 10, et\ k < n$) . On vous demande d'écrire un algorithme qui tri le tableau A, sans le manipuler directement, mais à travers la procédure $Sort$ seulement.

Exercice 5 (5pts)

On nous donne n liquides $L_1, L_2, \dots, L_n$. Chaque liquide L_i à un volume v_i et un poids p_i. On nous donne aussi un récipient de capacité V. Il est demandé de remplir le récipient au maximum possible tout en minimisant son poids (x_i est la fraction du volume v_i du liquide L_i utilisé). L'algorithme suivant est utilisé pour résoudre le problème. L'algorithme est-il un algorithme glouton ? Expliquez.

```
Trier les liquides par ordre non décroissant des densités pi/vi ;
pi/vi ≤ pi+1/v+1i

v = V ; i = 1 ;
While (vi ≤ V)
  { xi = 1 ;
    v = v − vi ;
    i = i + 1 ;
  }
xi = v/vi ;
for (j = i + 1; j ≤ n; j + +) xj = 0 ;
}
```

Exercice 6 (10pts)

Sur le graphe représenté sur la figure.1, nous avons placé des jetons dans des nœuds choisis de manière arbitraire (un jeton au plus par nœud). On veut connaitre le nombre maximum de jetons qu'on peut collecter en parcourant le graphe depuis le nœud central (nœud 0) jusqu'à un nœud externe n donné.

a) Donner le relation de récurrence qui va servir de base à l'écriture d'un algorithme récursif résolvant ce problème. Expliquer comment aboutir à cette relation.
b) Ecrire l'algorithme récursif (programme principal et fonction de calcul).
c) Expliquer le problème qu'il y a avec cet algorithme, et dire comment le résoudre ?
d) On relie les nœuds externes du graphe dans le sens contraire du déplacement des aiguilles d'une montre. Modifier l'algorithme pour qu'il réponde au même problème posé.

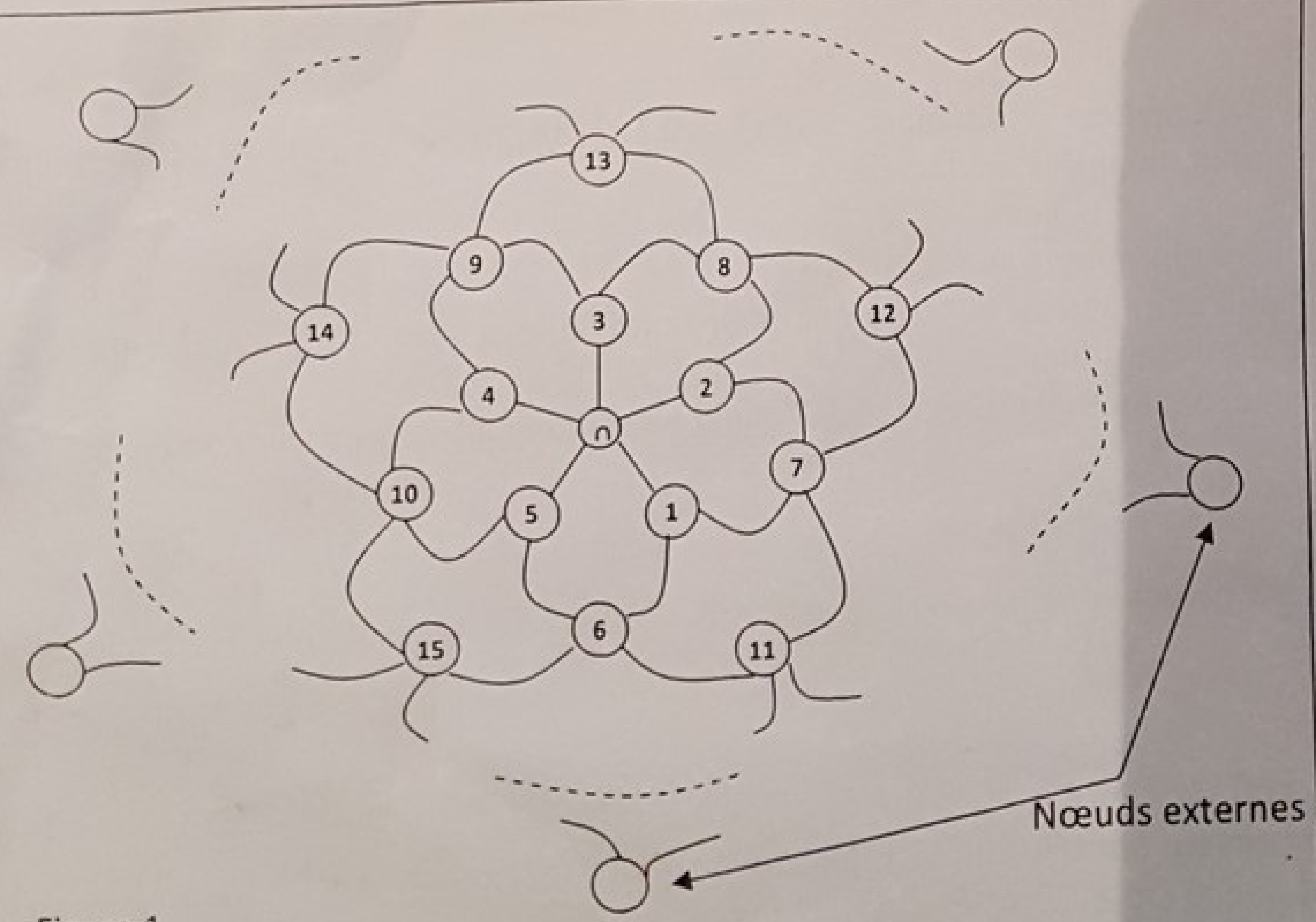


Figure.1