

M2R - NSI - Optimisation et compilation

20 février 2012

Barème indicatif: problème 1: 40%, problème 2: 40%, problème 3: 20%. Les 3 problèmes sont indépendants. Les questions n'ont pas forcément une réponse unique. Les analyses critiques seront appréciées. La note finale prendra en compte cet examen (coef 4) et le résumé d'article demandé par M. Bachir (coef 1).

1 Problème 1

On considère la boucle bi-dimensionnelle ci-dessous qui produit la matrice G à partir de la matrice S :

```
for(i = 0; i < n; i++)
  for(j = i; j < 2n - i; j++)
    G(i, j) = max(S(i, 2*i) , S(j - i, j)) + S(i, 2*(j - i));
```

1. Écrire l'espace d'itération sous forme polyédrique (i.e. $Ax \leq b$). Expliciter mathématiquement A et b , et écrire les 4 fonctions d'accès.
2. Montrer que les fonctions d'accès en S sont toutes valides en supposant que G et S sont des matrices $n \times 2n$.
3. Calculer le volume de l'espace d'itération.
4. Peut-on simplement intervertir les deux boucles sur i et j ? Pourquoi ?
5. Écrire une version équivalente du code avec les boucles en i et en j inversées.
6. En supposant que les points de G calculés par la boucle sont les seuls utiles, proposer une linéarisation $\ell(i, j)$ qui réalise un stockage compact de G .
7. Définir et expliquer deux types de transformations de boucles au choix.

2 Problème 2

A, B, C, \dots sont les variables d'un programme. Les opérations du langage sont toutes les opérations n -aires (à n opérandes) formées de $+$ et de $-$. Par exemple $D := A - B + C$ est une telle opération ternaire.

On considère la séquence d'affectations:

$A := A + B;$

$B := A - B;$

$A := A - B;$

1. Montrer que cette séquence réalise l'échange des valeurs contenues dans A et B ; quel est l'intérêt d'utiliser cette séquence?
2. Ce programme est-il réversible avec des opération binaires (à 2 opérandes)? Justifier.
3. En utilisant des opérations binaires, écrire une séquence d'opérations qui réalise la permutation circulaire des valeurs contenues dans les variables A, B et C et utilise seulement ces 3 variables: à la fin du programme, B contient la valeur initiale de A , C contient la valeur initiale de B , A contient la valeur initiale de C . Noter le nombre d'opérations.
4. En utilisant des opérations binaires et/ou ternaires, montrer qu'on peut réaliser la permutation en 4 opérations en utilisant seulement les 3 variables A, B et C .
5. Ce programme est-il réversible? Justifier.
6. Montrer qu'on peut réaliser la permutation circulaire de n variables A_1, A_2, \dots, A_n en n'utilisant pas d'autre variable, avec une séquence réversible de $n + 1$ opération n -aires.
7. Quel est l'intérêt d'utiliser cette séquence? Commenter.

3 Problème 3

On considère un ensemble de processeurs ayant des canaux de communications directs entre eux. Le graphe dont les noeuds sont les processeurs et les arêtes sont les canaux représente l'architecture du réseau.

1. Donner trois exemples simples de graphes pour lesquels les contraintes liées a la prise en compte de la localité de l'espace sont respectées.
2. Pour chacun des ces trois exemples, donner la complexité en temps de l'algorithme *quicksort* vu en cours, en supposant que le pivot sépare à chaque fois les données à trier en deux groupes de même taille.

Les exemples doivent être choisis de façon à ce que cette complexité ne soit pas la même pour chacun des trois cas.