7585-A High Performance Computing
Problem 2.1

Full Name: Ramesh Adhikari
ID Number: 912147172

February 2023

```
for ( i = 0; i < MAX; i++ )
 for ( j = 0; j < MAX; j++ )
  for ( k = 0; k < MAX; k++ )
   for ( m = 0; m < MAX; m++)
    BODY(i, j, k, m) ;
```

BODY .

(a) A[i, j, k, m] += A[i-1, j, k, m];
R[m] += A[i+1, j-1, k,m];

(b) R[m] += A[i-1, j-1, k+1, m+1];
A[i, j, k, m] += A[i-1, j+1, k, m];

Task: Draw dependence graph for these two cases, and determine all legal transformations

Solution (a)
Lets us find out the distance vector and direction vector of the body element of the loop. An interchange (transformation) is legal if positive direction vectors remain positive and negative remains negative.

Body	Distance Vector	Direction Vector
A[i, j, k, m] += A[i-1, j, k, m]	[1, 0, 0, 0]	[+, 0, 0, 0]
R[m] += A[i+1, j-1, k, m]	[-1, 1, 0, 0]	[-, +, 0, 0]

Let us analyze the loop

S1: A[i, j, k, m]	= A[i, j, k, m] + A[i-1, j, k, m];
S2: R[m]	= R[m] + A[i+1, j-1, k, m];
S1(1,1,1,1): A[1, 1, 1, 1]	= A[1, 1, 1, 1] + A[0, 1, 1, 1]
S2(1,1,1,1): R[1]	= R[1] + A[2, 0, 1, 1]
S1(1,1,1,2): A[1, 1, 1, 2]	= A[1, 1, 1, 2] + A[0, 1, 1, 2]
S2(1,1,1,2): R[2]	= R[2] + A[2, 0, 1, 2]
S1(1,1,1,3): A[1, 1, 1, 3]	= A[1, 1, 1, 3] + A[0, 1, 1, 3]
S2(1,1,1,3): R[3]	= R[3] + A[2, 0, 1, 3]
S1(1,1,2,2): A[1, 1, 2, 2]	= A[1, 1, 2, 2] + A[0, 1, 2, 2]
S2(1,1,2,2): R[2]	= R[2] + A[2, 0, 2, 2]
S1(1,2,1,1): A[1, 2, 1, 1]	= A[1, 2, 1, 1] + A[0, 2, 1, 1]
S2(1,2,1,1): R[1]	= R[1] + A[2, 1, 1, 1]
S1(2,1,1,1): A[2, 1, 1, 1]	= A[2, 1, 1, 1] + A[1, 1, 1, 1]
S2(2,1,1,1): R[1]	= R[1] + A[3, 0, 1, 1]
S1(2,2,1,1): A[2, 2, 1, 1]	= A[2, 2, 1, 1] + A[1, 2, 1, 1]
S2(2,2,1,1): R[1]	= R[1] + A[3, 1, 1, 1]

Table 1: Sample iteration over loop

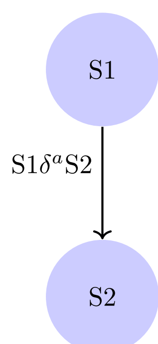


Figure 1: Anti-dependence between S1 and S2. In the statement, S2 is reading the value of A which will be updated by S1 in the future iteration. Details of the iteration are shown below the Table 1.

Legal Transformation:
If we start with i then we will have a negative direction vector. Therefore legal transformations are the below permutation list:
jikm, jimk, jkim, jkmi, jmik, jmki, kjim, kjmi kmji, mjik, mjki, mkji

Solution (b)
For This problem also, lets us find out the distance vector and direction vector of the body element of the loop. An interchange (transformation) is legal if positive direction vectors remain positive and negative remains negative.

Body	Distance Vector	Direction Vector
R[m] += A[i-1, j-1, k+1, m+1]	[1, 1, -1, -1]	[+, +, -, -]
A[i, j, k, m] += A[i-1, j+1, k, m]	[1, -1, 0, 0]	[+, -, 0, 0]

Let us analyze the loop

S1: R[m]	= R[m] + A[i-1, j-1, k+1, m+1];
S2: A[i, j, k, m]	= A[i, j, k, m] + A[i-1, j+1, k, m];
S1(1,1,1,1): R[1]	= R[1] + A[0, 0, 2, 2]
S2(1,1,1,1): A[1, 1, 1, 1]	= A[1,1,1,1] + A[0, 2, 1, 1]
S1(1,1,1,2): R[2]	= R[2] + A[0, 0, 2, 3]
S2(1,1,1,2): A[1, 1, 1, 2]	= A[1,1,1,2] + A[0, 2, 1, 2]
S1(1,1,2,1): R[1]	= R[1] + A[0, 0, 3, 2]
S2(1,1,2,1): A[1, 1, 2, 1]	= A[1,1,1,1] + A[0, 2, 2, 1]
S1(1,1,2,2): R[2]	= R[2] + A[0, 0, 3, 3]
S2(1,1,2,2): A[1, 1, 2, 2]	= A[1,1,1,1] + A[0, 2, 2, 2]
S1(2,1,1,1): R[1]	= R[1] + A[1, 0, 2, 2]
S2(2,1,1,1): A[2,1,1,1]	= A[2,1,1,1] + A[1, 3, 1, 1]
S1(2,2,1,1): R[1]	= R[1] + A[1, 1, 2, 2]
S2(2,2,1,1): A[2,2,1,1]	= A[2,2,1,1] + A[1, 3, 1, 1]
S1(2,2,2,2): R[2]	= R[2] + A[1, 1, 3, 3]
S2(2,2,2,2): A[2,2,2,2]	= A[2,2,2,2] + A[1, 3, 2, 2]
S1(3,3,1,1): R[1]	= R[1] + A[2, 2, 2, 2]
S2(3,3,1,1): A[3,3,1,1]	= A[3,3,1,1] + A[1, 3, 2, 2]

Table 2: Sample iteration over loop

Legal Transformation:
If we start with j,k,m then we will have a negative direction vector. Therefore legal transformations are the below permutation list which is stated with i.
ijkm, ijmk, ikjm, ikmj, imjk, imkj

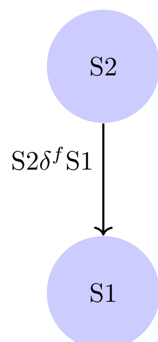


Figure 2: Flow-dependence between S1 and S2. In the statement, S1 is reading the value of A which is updated by S2 in the earlier iteration. Details of the iteration are shown in Table 2.

(a) solution of A

A[i, j, k, m] = A[i, j, k, m] + A[i-1, j, k, m]

R[m] = R[m] + A[i+1, j-1, k, m]

indexex
i, j, k, m

1, 1, 1, 1 A[1,1,1,1] = A[1,1,1,1] + A[0, 1, 1, 1]

R[1] = R[1] + A[2, 0, 1, 1]

i, j, k, m

1, 1, 1, 2 A[1, 1, 1, 2] = A[1, 1, 1, 2] + A[0, 1, 1, 2]

R[2] = R[2] + A[2, 0, 1, 2]

1, 1, 2, 2 A[1, 1, 2, 2] = A[1, 1, 2, 2] + A[0, 1, 2, 2]

R[2] = R[2] + A[2, 0, 2, 2]

1, 2, 1, 1 S_1 A[1, 2, 1, 1] = A[1, 2, 1, 1] + A[0, 2, 1, 1]

S_2 R[1] = R[1] + A[2, 1, 1, 1]

anti-flow

2, 1, 1, 1 S_1 A[2, 1, 1, 1] = A[2, 1, 1, 1] + A[1, 1, 1, 1]

S_2 R[1] = R[1] + A[3, 0, 1, 1]

2, 2, 1, 1 S_1 A[2, 2, 1, 1] = A[2, 2, 1, 1] + A[1, 2, 1, 1]

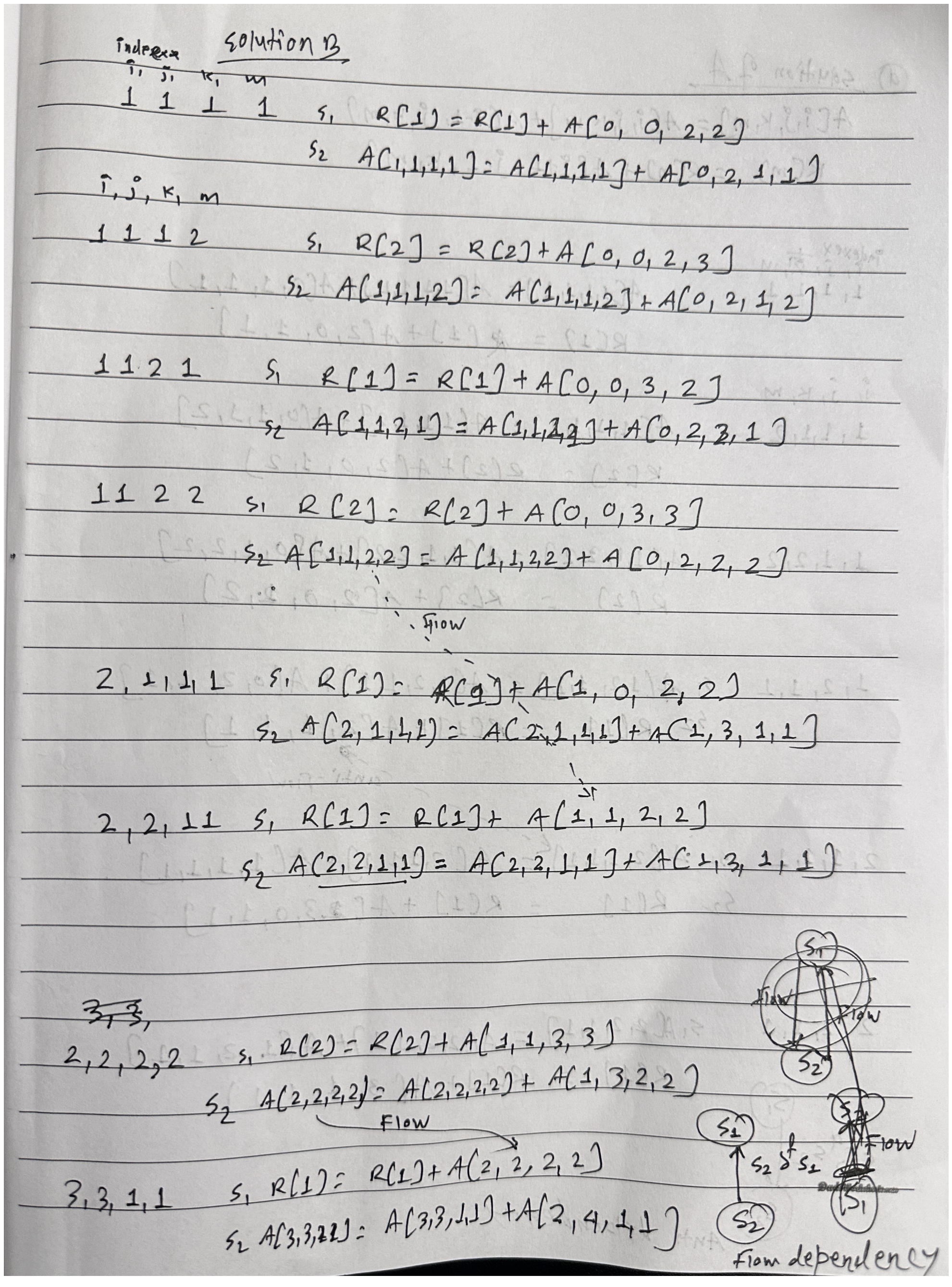
R[1] = R[1] + A[3, 1, 1, 1]

$S_1 \delta^a S_2$

$S_1 \rightarrow S_2$

Anti dependency

Figure 3: Analyzation of the dependency for Question A



Indexes Solution B

i, j, k, m

1 1 1 1 S_1 R[1] = R[1] + A[0, 0, 2, 2]

S_2 A[1,1,1,1] = A[1,1,1,1] + A[0, 2, 1, 1]

i, j, k, m

1 1 1 2 S_1 R[2] = R[2] + A[0, 0, 2, 3]

S_2 A[1,1,1,2] = A[1,1,1,2] + A[0, 2, 1, 2]

1 1 2 1 S_1 R[1] = R[1] + A[0, 0, 3, 2]

S_2 A[1,1,2,1] = A[1,1,2,1] + A[0, 2, 3, 1]

1 1 2 2 S_1 R[2] = R[2] + A[0, 0, 3, 3]

S_2 A[1,1,2,2] = A[1,1,2,2] + A[0, 2, 2, 2]

Flow

2, 1, 1, 1 S_1 R[1] = R[1] + A[1, 0, 2, 2]

S_2 A[2,1,1,1] = A[2,1,1,1] + A[1, 3, 1, 1]

2, 2, 1, 1 S_1 R[1] = R[1] + A[1, 1, 2, 2]

S_2 A[2,2,1,1] = A[2,2,1,1] + A[1, 3, 1, 1]

~~3, 3,~~

2, 2, 2, 2 S_1 R[2] = R[2] + A[1, 1, 3, 3]

S_2 A[2,2,2,2] = A[2,2,2,2] + A[1, 3, 2, 2]

Flow

3, 3, 1, 1 S_1 R[1] = R[1] + A[2, 2, 2, 2]

S_2 A[3,3,1,1] = A[3,3,1,1] + A[2, 4, 1, 1]

S_1 Flow Flow S_2

S_1 $S_2 \, \delta^f \, S_1$ S_2 Flow S_1

Flow dependency

Figure 4: Analyzation of the dependency for Question B