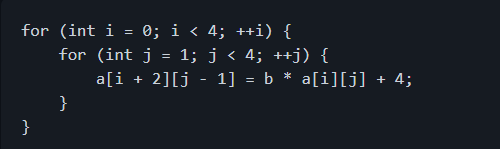
**Task 3)**

****

* For each iteration, give the corresponding distance and direction vectors

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | **Source** | **Sink** |  |  |
| **I** | **J** | **S1: a[i+2][j-1]** | **S1: a[i][j]** | **Distance** | **Direction Vec** |
| 0 | 1 | a[2][0]= | =a[0][1] | (2,-1) | (<,>) |
| 0 | 2 | a[2][1]= | =a[0][2] | (2,-1) | (<,>) |
| 0 | 3 | a[2][2]= | =a[0][3] | (2,-1) | (<,>) |
| 1 | 1 | a[3][0]= | =a[1][1] | (2,-1) | (<,>) |
| 1 | 2 | a[3][1]= | =a[1][2] | (2,-1) | (<,>) |
| 1 | 3 | a[3][2]= | =a[1][3] | (2,-1) | (<,>) |
| 2 | 1 | a[4][0]= | =a[2][1] | (2,-1) | (<,>) |
| 2 | 2 | a[4][1]= | =a[2][2] | (2,-1) | (<,>) |
| 2 | 3 | a[4][2]= | =a[2][3] | (2,-1) | (<,>) |
| 3 | 1 | a[5][0]= | =a[3][1] | (2,-1) | (<,>) |
| 3 | 2 | a[5][1]= | =a[3][2] | (2,-1) | (<,>) |
| 3 | 3 | a[5][2]= | =a[3][3] | (2,-1) | (<,>) |

* What type of dependence is represented in this code snippet?

This is a true dependence, because we write first into an index and read in later iterations.

For example: In iteration i=0, j=2, we write into a[2][1] and in iteration i=2,j=1 we read a[2][1]

This dependence is loop carried and can not be parallelized just like that (#pragma omp parallel for)

* How would you parallelize this code?

You can parallelize this snippet with DOACROSS loops:

DOACROSS I = 0, 4

DO J = 1, 4

IF I .NE. 0 WAIT(EV(J), lNeighb)

a[I+2][ J-1] = b \* a[i][j] + 4

IF I .NE. 4 POST (EV(J), rNeighb)

ENDDO

ENDDO

Where lNeighb ist he left neighbour thread and rNeighb the right neighbour thread.

NE stands for “not equal”.