

CS 6230/5960

Homework 3

Due 11:59pm on Friday, 2/21/2020 (No late submissions)

1. (20 points) Consider the following code:

```
for (t=0; t<1024; t++)
  for (i=0; i<1024; i++)
    for (j=0; j<i; j++)
      for (k=0; k<i; k++)
        S(t, i, j, k);
```

The data dependences for the loop are given to be (0,0,1,-1), (1,-2,1,-1), and (1,-1,1,0).

- (a) Which loops (if any) are valid to unroll?
 - (b) What are the valid permutations of the loop?
 - (c) What tiling (if any) is valid?
2. (20 points) Consider the following computation:

```
double A[1024][1024]; int t, i, j;
for (t = 0; t < 1024; t++)
  for (i = 1; i < 1023; i++)
    for (j = 1; j < 1023; j++)
      A[i+1][j-1] = A[i][j+1] + A[i-1][j];
```

- (a) Are there any output dependences? If so, state the output dependence vector (if there are many possible such vectors, list the one with the smallest magnitude). You must show why the dependence vector is valid (either via algebraic derivation or by some enumeration).
 - (b) Repeat above for flow and anti dependences.
3. (20 points) For the following loop code: i) Determine loop-carried dependences; ii) Find the level of dependences and identify any parallel loops; iii) Attempt to increase the total number of parallel loops by performing loop permutation (if it cannot be increased, explain why).

```
for (i=1; i<256; i++)
  for (j=1; j<256; j++)
    for (k=1; k<256; k++)
      { a[i][j][k] = b[i][j][k-1]+1;
        b[i][j][k] = c[i][j-1][k]-1;
        c[i][j][k] = a[i-1][j][k]+1;
      }
```

4. (20 points) For the following multi-statement loop code, i) Form the statement-level dependence graph, marking dependence edges as flow or anti dependences; ii) Generate maximally vectorized code (performing suitable loop transformations as needed).

```
for (i=1; i<256; i++)
{ a[i] = c[i]+1;
  b[i+1] = a[i]-1;
  c[i+1] = a[i-1]+1;
}
```

5. (20 points) For the following nested loop code, i) Determine data dependences; ii) Show vectorized code (performing suitable loop transformations as needed); if it cannot be vectorized, explain why.

```
for (i=1; i<256; i++)
  for (j=1; j<256; j++)
    for (k=1; k<256; k++)
      a[i][j][k] = a[i-1][j+1][k+1]+1;
```