# Department of Computer Science & Engineering
# CSL 728 Compiler Design
# Major Test

Date: 05.05.2010 Time: 08:00 – 10:00 Max. Marks: 25

1. How would you eliminate copy operations like MOVE ($temp_i$ , $temp_j$) during graph coloring based register allocation, where $temp_i$ and $temp_j$ are compiler generated temporaries. Clearly state the conditions you would check before doing this elimination. How is this related to copy propagation optimization?

   (5)

2. Write the procedure for run time garbage collection using copying. How is it better or worse than mark and sweep approach?

   (4)

3. Can the following approach work for hoisting loop invariant operation out of a loop? Create a pre-header of the loop and introduce a copy of the candidate operation in it. Then apply common sub-expression elimination optimization to see if the original operation in the loop body can be eliminated.

   (4)

4. Write an algorithm to identify loops and to construct loop nest tree.

   (4)

5. Illustrate software pipelining using a simple loop example.

   (3)

6. Consider the following program for transposing an $n \times n$ matrix. How would you restructure the program so that a very large matrix may be efficiently transposed on a system with a relatively small cache.

```
for i = 1 to n - 1 do
   for j = 0 to i - 1 do
      temp = A [i, j]
      A [i, j] = A [j, i]
      A [j, i] = temp
   end for
end for
```

   (5)