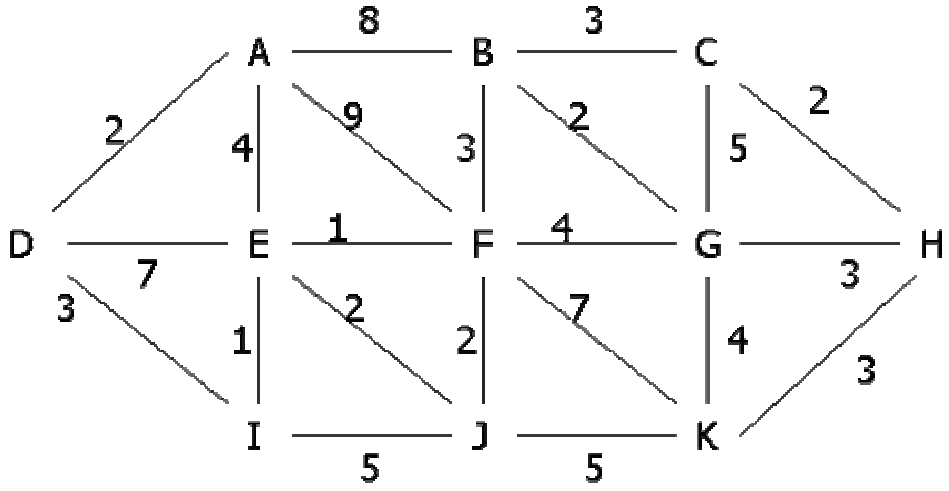
B. Math III Year
2005 Final Examination – CS III Algorithms and Data Structures
(Max. marks: 50 and Time 3 hours)

Note: Don't write long essays. Instead, spend more time in thinking about the questions and then give brief answers. Good luck!

Q1 [5 Marks: 1 Mark for each part]

- **a.** If f(n)+g(n) is $O(n^2)$ then f(n) is $O(n^2)$ (True/False)
- **b.** If f(n) is $O(n^2)$ and g(n) is $O(n^2)$ then f(n)/g(n) is O(1) (True/False)
- **c.** Prim's and Kruskal's algorithms will always give the same minimum spanning tree for a given connected graph (True/False)
- **d.** Select the correct words from the brackets: With MergeSort, the data comparisons are all done (after/before) the recursive calls complete, whereas with QuickSort the data comparisons are all done (after/before) the recursive calls are made.
- **e.** Given a binary tree with integer data, an inorder traversal of the tree will output the data in sorted order. (True/False)

Q2 [15 Marks] Solve one of the following two:
a. Use Prim's algorithm to find a minimum spanning tree (MST). Now use Kruskal's algorithm to find the MST. Show your reasoning and if you make any assumptions then state them clearly.



b. Obtain a closed form expression for the order of the average time taken for a quicksort algorithm. If you make any assumption about the algorithm then state them clearly.

B. Math III Year
2005 Final Examination – CS III Algorithms and Data Structures
(Max. marks: 50 and Time 3 hours)

Q3 [30 Marks] These days most newspapers feature the number game Sudoku. For those who are not familiar with this puzzle there is a sample given below. It is a game that one solves using reasoning and logic. The missing numbers are filled in the grid in such a way that every row, every column, and every 3x3 box accommodates the digits 1 to 9 without repeating any. Thus given a grid in figure 1, the aim is to fill it so that you get the grid in figure 2. A given Sudoku grid has (to have) a unique solution.

6		**5**			**2**	**3**		
			3	**5**		**2**		
	2	**3**					**1**	
4	**7**		**2**				**3**	
			4		**3**			
	3				**7**		**4**	**1**
	8					**5**	**2**	
		1		**6**	**4**			
		4	**7**			**1**		**6**

figure 1

6	1	**5**	9	4	**2**	**3**	8	7
8	4	7	**3**	**5**	1	**2**	6	9
9	**2**	**3**	8	7	6	4	**1**	5
4	**7**	9	**2**	1	5	6	**3**	8
1	6	8	**4**	9	**3**	7	5	2
5	**3**	2	6	8	**7**	9	**4**	**1**
7	**8**	6	1	3	9	**5**	**2**	4
2	9	**1**	5	**6**	**4**	8	7	3
3	5	**4**	**7**	2	8	**1**	9	**6**

figure 2

Sudoku grids present two interesting problems to the students of algorithms and data structures. The first – how to create a Sudoku puzzle and the second - how to solve it. Sub-questions 1 to 5 focus on the first problem and the remaining on the second:

1. What according to you determines the degree of difficulty of the created grid? [**3 marks**]
2. In about 15-20 lines describe the logic (**not the algorithm**!) that you will use to create a grid like the one in figure 1 for a given level of a difficulty and a given size of the grid. [**5 marks**]
3. Now create an algorithm using the logic you have specified earlier. [**10 marks**]
4. Clearly state all of the data structures you have used in your algorithm. [**2 marks**]
5. If *n* is the total number of digits used in the grid or one of the dimensions of the square grid (9 in the example above) then what is the order of your algorithm in terms of *n*? [**3 marks**]
6. Given a grid like the one in figure 1 it is fairly easy to create a program to solve it. One can take several approaches so pick the one that you think is efficient and describe the logic behind it in about 15-20 lines [**Hint**: if you are thinking of a brute force method where you will try every number in every empty location, well it is not efficient so think again!] [**7 marks**]

***Note**: Marks will be awarded with a due consideration to the quality of your algorithm, adherence to the pseudo-code conventions, brevity/cogency of your answer and that of the approach.*