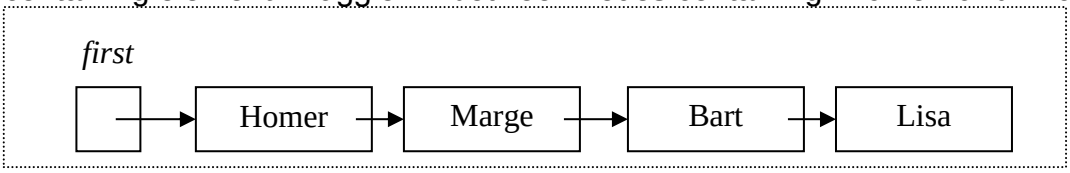
Workshop Test 2: Array sorting, LL and binary trees (your marks: (out of 10))

Student ID: **Name:** .

1. (2 marks)
   a) A binary tree has 600 nodes. What is the maximum possible depth of the tree? And what is the minimum possible depth of the tree?
   b) List the key properties (or features) of a binary search tree.

2. (2 marks)
   The two elementary array sorting algorithms, *Selection* and *Insertion,* have the same time complexity of $O(n^2)$ on average case. However, their performances, in terms of number of comparisons, can be quite different in some special cases even on average cases. Compare the two algorithms and determine
   a) If A[ ] is already sorted, which of the above-mentioned sorting algorithms may achieve an O(*n*) time complexity to complete the sorting procedure? Explain your answer using an example.
   b) Which algorithm is better to sort A[0…n-1], in general case, in term of the number of comparisons? Why?

3. (1 mark)
   Given the following linked list, write one Java-like sentence/s to insert a node containing element "Maggie" in between nodes containing "Homer" and "Marge".



4. (3 marks)
   a) Insert the following animals into an empty BST (you may show the final BST only):

      goat, pig, dog, cow, rat, cat, tiger, fox, lion

   b) After the insertions, show the results of *pre-order*, *in-order*, and *post-order* traversals of the BST.

5. (2 marks)
   An array *A*[0 … *n-1*], where *n* > 10, stores the result of assignment 1 of this unit. Assume that all marks are integers, and the array is unsorted. By a preliminary rule, all students who achieved a mark on the top-10 mark-list of the class will be qualified to receive an award.
   a) Write an algorithm, *print_top10(A, n),* that prints the 10 top marks of the class.
   b) Analyse your algorithm using *O*-notation.

END OF THE TEST 02