

CSE 150: Problem Set #4

December 3, 2007

Problem 1

Write two recursive functions that operate on linked lists:

- `insert(L,x)`. If `L` is sorted, then `insert(L,x)` returns a sorted linked list containing all the elements of `L`, and `x`. For example, `insert([1,3,4], 2) = [1,2,3,4]` and `insert([1,2,3], 1) = [1,1,2,3]`.
- `sort(L)` returns a sorted version of `L`. For example, `sort([2,1,3,0]) = [0,1,2,3]` and `sort([])=[]`.

Recall that the basic functions you can use are `head()`, `tail()`, `cons()`, and `isempty()`.

What are the running times of your algorithms?

Problem 2

Let T be a full binary tree (i.e. a binary tree in which each node has 0 or 2 children) with n leaves. How many internal nodes are there? Prove your answer.

Problem 3

How many different undirected graphs are there on n nodes? Directed graphs?

Problem 4

Examine the Bridges of Königsberg problem in Figure 1, where white boxes are “islands” and grey boxes are “bridges”. Is it possible to walk around Königsberg and walk over each bridge exactly once? Why or why not?

An Eulerian path in a graph is one which crosses every edge exactly once. An Eulerian cycle is an Eulerian path which starts and ends on the same vertex. Is there an Eulerian path on the Königsberg graph (Figure 1)? Why not? **Prove that a connected graph G contains an Eulerian cycle if and only if every vertex of G has even degree.**

Problem 5

Prove that the following algorithm outputs some permutations more often than others:

```
procedure genPerm(n)
  A := [ 1, 2, 3, ..., n ]
  for i = 1 to n
    j := random number between 1 and n
    k := random number between 1 and n
    SWAP(A[j], A[k])
  return A
```

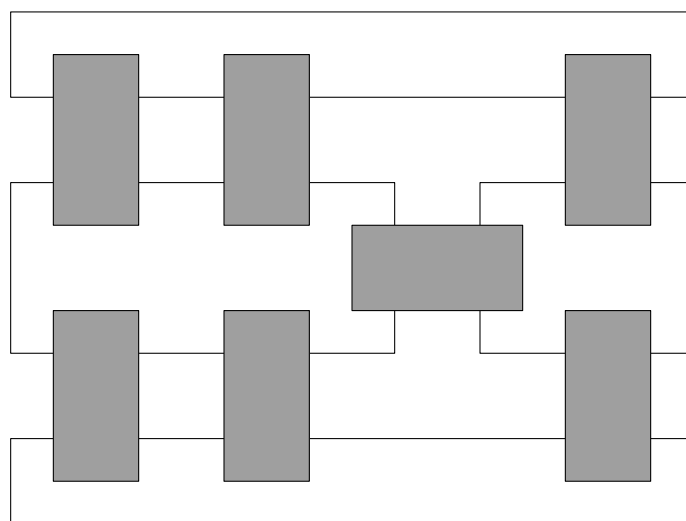


Figure 1: The Bridges of Königsberg