Algorithms – CS-37000 Homework 3 – January 10, 2006

Instructor: László Babai Ry-164 e-mail: laci@cs.uchicago.edu

ADVICE. Take advantage of the TA's office hours.

READING KT, material about BFS. "Knapsack" handout.

REVIEW from Discrete Math: Graphs and Digraphs, Asymptotic notation. Review all previous handouts and readings.

HOMEWORK. Please **print your name on each sheet.** Put every solution on a separate sheet. Please try to make your solutions easily readable.

This homework is due on **Tuesday, January 17** at the **beginning of the class.**

In the problems below, the digraph G is given by an *array of adjacency lists* ("adjacency list representation" for short). We assume that there are no multiple items on these lists. An algorithm runs in *linear time* if its cost is big-Oh of the length of the input, so it is $O(n+m)$ under our tacit assumption. Unit cost is associated with the following operations: looking up and copying an integer between 1 and n, looking up an entry at a given location in an array, moving to the next item in a linked list, and similar basic bookkeeping operations.

Unless expressly stated otherwise, **all algorithms must be described in pseudocode**. IMPORTANT: **Define and explain your variables** and put comments on the lines of your pseudocode. (Otherwise your code will be unintelligible. You lose credit if understanding your solution requires unreasonable amount of work.)

All problems below have VERY SIMPLE solutions (just a few lines of pseudocode). ELEGANCE COUNTS!

For each problem below, state your answer in the form of an elegant pseudocode. Add definitions of your variables and brief comments explaining what is happening. No explanation of the correctness and the cost of your algorithms is required (as long as they are indeed correct and run in linear time).

3.1 (0 points, do not hand in) Find the connected components of an undirected graph. Write a pseudocode for the algorithm outlined in class.

3.2 (5 points) Recall that a digraph is *strongly connected* if each vertex is accessible from every vertex. Given a digraph (in adjacency list representation), decide in linear time whether or not it is strongly connected. (Hint: a simple combination of previously studied algorithms.)

3.3 (12 points) *(All-ones square problem.)* Given an $n \times n$ array A of zeros and ones, find the maximum size of a contiguous square of all ones. (You do not need to locate such a largest all-ones square, just determine its size.) Solve this problem in *linear time*. "Linear time" means

the number of steps must be O(size of the input). In the present problem, the size of the input is $O(n^2)$. Manipulating integers between 0 and n counts as one step; such manipulation includes copying, incrementing, addition and subtraction, looking up an entry in an $n \times n$ array.

Describe your solution in **pseudocode.** Make sure you give a clear **definition of the variables you introduce**; this accounts for half the credit. The solution should be *very simple,* no more than a few lines. **Elegance counts.** *Hint:* dynamic programming. Example:

1	0	1	1	0	1
1	1	0	1	1	1
1	0	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	0
1	1	1	0	1	1

In this example, the answer is 3. There are three contiguous 3×3 square subarrays of all-ones. One is indicated below by underlines, another is shown in a box, the third one is indicated by *Italics.*

1	0	1	1	0	1
1	1	0	*1*	*1*	*1*
1	0	1	*1*	*1*	*1*
1	1	1	*1*	*1*	*1*
1	1	1	1	1	0
1	1	1	0	1	1

3.4 (14 points) Consider the Knapsack problem described in the handout. The input parameters in that problem are positive reals called *weights* and *values*; and a "weight limit" W is given. It is shown in the handout how one can solve this problem in $O(nW)$ steps (arithmetic, comparison, bookkeeping) if all *weights* (including W) are *integers.*

Assume now that all *values* are *integers* (but the weights are real). Let V denote the sum of the values. Solve the knapsack problem in $O(nV)$ steps under this assumption. Your solution should be a simple **pseudocode**. Make sure you give a clear **definition of your variables;** this accounts for half the credit.