Eksamination in course
SIF8010 Algorithms and Data Structures
Friday August 9 2002, 0900-1500

Contact during the examination: Arne Halaas, ph. 73593442.
Tools: All calculator types allowed. All printed and handwritten material allowed.
Important: You should preferrably write your answers only on the separate answer form. If necessary, extra sheets may be submitted. The problems are meant to be answered succinctly.

Problem 1

Consider the directed graph *G*=(*V*, *E*) given by the following edge weight matrix:

$$W = \begin{bmatrix} \infty & 23 & 157 & 60 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & 166 & \infty & \infty \\ \infty & 187 & \infty & 195 & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 80 & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & 176 \\ \infty & \infty & \infty & \infty & \infty & \infty & 173 \\ \infty & \infty & 183 & \infty & \infty & \infty & \infty \end{bmatrix}$$

From this matrix we can, for instance, see that there is an edge from node 1 to node 2 with weight 23, because the element $w_{12} = 23$ (where w_{ij} is the element in row *i* and column *j*). However, there is no edge from node 3 to node 6, because $w_{36} = \infty$.

Assume that the following rule holds: If an algorithm may choose between several equivalent alternatives, it will choose the node with the lowest numeric value. For instance, if depth-first-search may visit node 4 or node 6, it will visit node 4.

a) In which order will the nodes be visited (for the first time) if a depth-first-search is performed, starting at node 1?

b) In which order will the nodes be visited (for the first time) if a breadth-first-search is performed, starting at node 1?

c) In which order will the nodes be added to the spanning tree if Prim's algorithm is performed, starting at node 1? Assume that the algorithm works on the underlying undirected graph with the same edge weights. For example, we have both $w_{32} = 187$ and $w_{23} = 187$.

d) In which order will the nodes be visited (for the first time) if Dijkstra's algorithm is performed, starting at node 1, on the original, directed graph?

Problem 2

a) Which of the following statements are true? (There may be none, one, or several true statements. If you believe none is true, simply leave the field in the aswer form empty.)

1) $n^{100} = O(2^n)$
2) $2^n = O(n^{100})$
3) $2^n = \Theta(3^n)$

b) Give a short explanation for each of the three statements in problem a). Why are they true/false?

c) Assume that you have a balanced binary search tree with *n* elements. You want to print out all the elements in the value range from *x* through *y*. Assume that there are *k* elements in this value range. Your algorithm must find all these elements and print them out. What is the running time complexity? Express the answer in theta notation.

d) What is the purpose of the heap structure? Why can't you simply use a binary tree (binary search tree)? One can use binary trees to find the largest (or smallest) element, and the average asymptotic costs for inserting and retrieving elements are the same. What, then, is the advantage of heaps over binary trees (for the specific application of heaps)?

Problem 3

Consider the following program in Java like pseudocode:

```
010  int mystery(int[] b, int[] c) {
020      int m = b.length
030      int n = c.length
040      int[][] a = new int[m][n]
050      initialize(a)
060      return f(b, c, m–1, n–1, a)
070  }
080  int f(int[] b, int[] c, int d, int e, int[][] a) {
090      if (d== –1 || e== –1)
100          return 0
110      else if (b[d]==c[e])
120          a[d][e] = f(b, c, d–1, e–1, a) + 1
130      else
140          a[d][e] = max(f(b, c, d–1, e, a), f(b, c, d, e–1, a))
150      return a[d][e]
160  }
```

The line numbering is used only in problem d). Assume that the function *initialize* sets all the elements in the array *a* to –1, and that the function *max*(*x*, *y*) returns the largest of the two numbers *x* and *y*.

Remember: Even if you may be uncertain about certain details in the program, you may be able to answer the questions.

a) The program implements an algorithm from the curriculum. Which one?

b) The running time of this simple recursive implementation is exponential. However, the problem being solved has a property that allows us to improve this running time substantially. Which property is this?

c) A well-known method from the curriculum can be used to improve recursive programs when the problem has the property described in problem b). The method lets the program remain recursive. What is the name of this method?

d) You are now going to improve the running time by adding a single line of code. Write this line of code (in Java or pseudocode) with the accompanying line number on the answer form. Give a line number that places the line correctly, for example, 075 to insert the line between line 070 and 080.