Algorithms and Complexity

Exam
9:00 - 12:00
September 21, 2004

Prof. Michael H. Böhlen

The exam consists of 6 exercises. The first exercise is a multiple choice exercise. Solve as many exercises as possible. It is important that you argue for your answers and that you present your solutions in a readable form. Fully specify all algorithms that you use as part of your solutions. During the exam no auxiliary material, except a simple pocket calculator, is permitted.

1 Multiple Choice Exercise (25 points)

1.1 Determine the time required by an algorithm to solve a problem of size $n = 10000$. Assume the algorithm requires $f(n)$ operations and processes 100 operations per second. Check the correct table cells.

$f(n)$	$<$ 0.01 sec	0.01 - 0.5 sec	0.5 - 30 sec	1 - 3 min	5 - 7 min	1 - 24 h	$>$ 10 days
$\log_{10} n$							
$\sqrt{n}$							
n							
$n \log_{10} n$							
n^2							

1.2 Insertion sort is applied to the array {31, 41, 59, 26, 41, 58}. What is the content of this array after 3 iterations of the outermost loop of insertion sort?

a) { 26, 31, 41, 41, 58, 59 }
b) { 26, 41, 59, 31, 41, 58 }
c) { 26, 31, 41, 59, 41, 58 }
d) { 26, 31, 41, 41, 59, 58 }

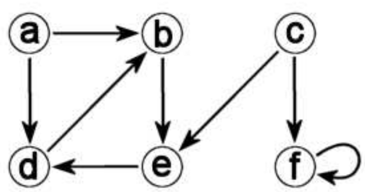
1.3 Which of the following equalities are true?

a) $n^k = O(c^n), k > 0, c > 1$
b) $2^n = O(2^{n/2})$
c) $2^n = \Omega(2^{n/3})$
d) $2^n = O(2^{2n})$
e) $n^2 = o(2n^2)$

1.4 A binary search tree was built by inserting the following numbers in the given order: 40, 20, 50, 10, 30, 60. Which of the following statements are true?

a) height of the tree is 3
b) the tree is a heap
c) 30 is the left child of 40
d) 50 is the right child of 40

1.5 Consider the graph below. BFS was run for this graph starting at vertex c.



What are the values of **dist** (distance) and **pred** (predecessor)for the nodes of the graph? Each pair below is in the form {**dist**,**pred**}, where the pairs correspond to vertices from a to f.
a) {∞,nil}, {4,d}, {0,nil}, {3,e}, {2,c}, {1,c}
b) {∞,nil}, {3,d}, {0,nil}, {2,e}, {1,c}, {1,c}
c) {∞,nil}, {3,d}, {0,nil}, {2,e}, {1,c}, {2,c}
d) {∞,nil}, {3,d}, {0,nil}, {2,e}, {1,f}, {1,c}
e) {4,d}, {3,d}, {0,nil}, {2,e}, {1,c}, {1,c}

2 Recurrence (20 points)

Consider the following recurrence:

$$T(1) = b$$
$$T(n) = bT(n-1) - (b-1) \quad (n \geq 2)$$

- What is the value of $T(n)$ for $n = 2, 3, 4$?

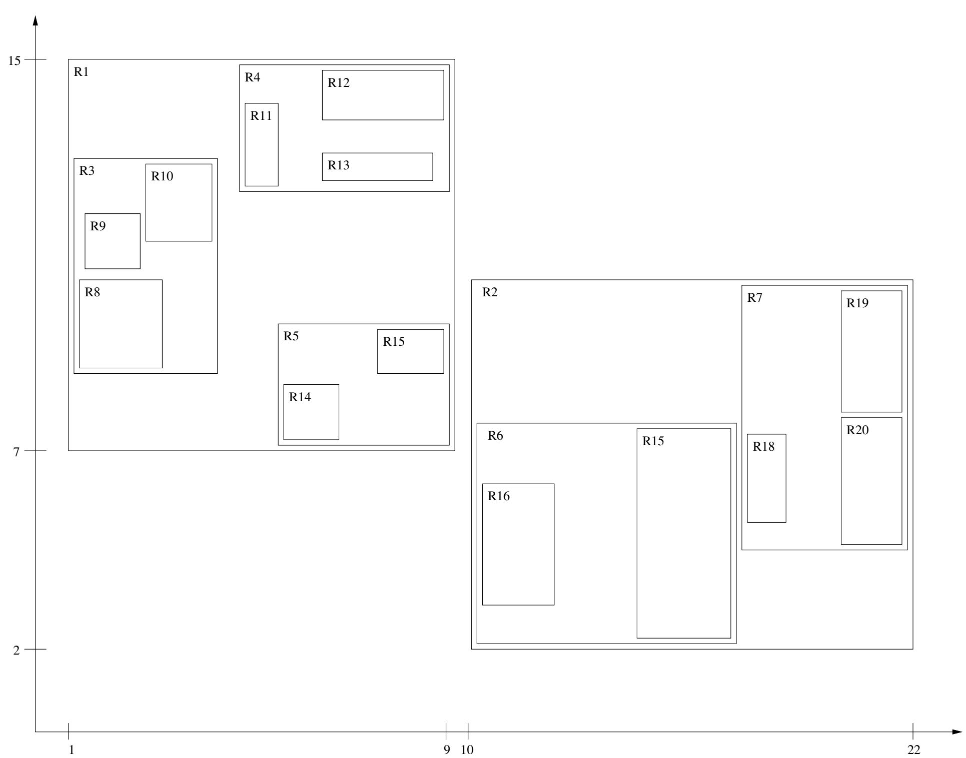
- Solve the above recurrence exactly by finding a closed form expression for it (without $T(n)$ on the right side of the expression).
- Find the simplest $f(n)$ such that $T(n) = \Theta(f(n))$.

3 Graphs (25 points)

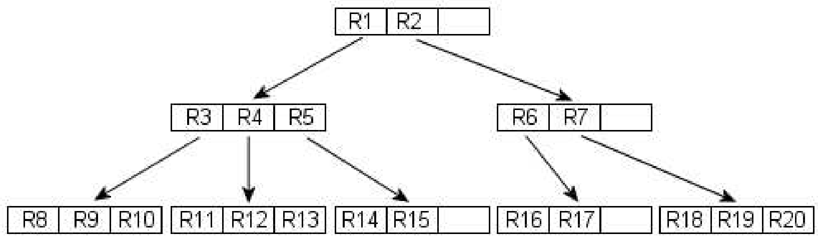
Let $G = (V, E)$ be a connected undirected graph. Give an $O(|V| + |E|)$-time algorithm to compute a path in G that traverses each edge in E exactly once in each direction. The output of your algorithm shall be the sequence of vertexes visited during the traversal.
Hint: Start out with the DFS algorithms. The original DFS traverses the vertices and colors them. In this exercise you have to traverse the edges.

4 Trees, C (30 points)

An R-Tree is a search tree for spatial data. The tree organizes rectangles in a hierarchy. Each rectangle covers an area that contains all the areas of the rectangles in its subtree. The following figure illustrates a hierarchy of depth 3. Rectangles R1 and R2 are at the top (first) level. Rectangles R3 - R7 are at the second level. Rectangles R8 - R20 are at the bottom (third) level



The figure below displays the exact same R-Tree as a search tree similar to the B tree. Each node holds a number of node entries. For example, the node at the top has two entries: R1 and R2. Each node entry represents a 2D rectangle with sides parallel to the axes. The rectangles are represented by the coordinates of two opposite corners.



Each node of the R-Tree contains up to m entries, where $m > 1$ is a fixed integer. Each node entry consists of a pointer to a child and a rectangle. Rectangles stored in nodes on the same level do not overlap.

1. Write a definition of R-Tree in C (a structure) where the constant m specifies the maximum number of entries in an R-Tree node. Assume that m is defined elsewhere. Use integers values for representing coordinates.

2. Write a function `bool contains(rtree* root, int x, int y)` that returns `true` if point (x, y) is contained in a rectangle of a leaf node and `false` otherwise.

   What is the worst case complexity of your function?

5 Recursion (30 points)

Rewrite the DFS algorithm for traversing a graph $G = (V, E)$. Your algorithm shall be non-recursive and use a stack. You may assume that vertexes are represented as integers numbers.
The set of stack operations that you may use is the following:

stack* StackInit() Creates and initializes a new empty stack and returns it.

StackPush(stack*, int) Puts given data at the top of the stack.

int StackPop() Removes an element from the top of the stack and returns it.

int StackSize(stack*) Returns number of elements in the stack.

What is the worst case complexity of non-recursive version of DFS?

6 Computer Network (25 points)

Assume a number of computers that are connected with network lines. Each network line directly connects two computers. Each computer is directly connected to a small number of other computers only. If a computer needs to send data to a computer that it is not directly connected to, the data is routed through other computers.

Our goal is to have the minimum necessary number of network lines, such that each computer can still send data to all other computers. For each network line we know its fault rate, which is defined as the number of data packets lost per day. The fault rate of a network is the sum of the fault rates of its network lines. Among the networks that have the minimum necessary number of network lines, we want a network that has the smallest fault rate. Such a network is called an efficient network.

Provide an algorithm that, for a given computer network, determines the network lines that must be left in the network to make it efficient (all other lines are removed). What is the worst case complexity of your algorithm?