

CS 324 Fall 2005

Name: _____

Test 3

Talking to other students and the use of notes, books, or electronic devices are not allowed and are considered cheating. There is a total of 85 points plus 20 points of extra credit, but one cannot get more than 85 points.

1. Algorithm Design Strategies (20 points)

a) Compare and contrast greedy algorithms, divide-and-conquer algorithms, and dynamic-programming algorithms. Give one example problem for each strategy, and explain why the problem can be solved using algorithms using that category.

2. Backtracking (15 points)

The following is the backtracking algorithm for solving the eight-queens problem. Draw that part of the implicit graph created by the algorithm that includes the first 10 nodes that are accessed and the edges between them. For each node, give the corresponding vector. The algorithm is started with the call `solve(0)`. At the beginning, all values in `grid[][]` are 0 (`grid[][]==0` means that the square is not threatened).

```
public boolean solve( int y ) {
    for( int i = 0; i < 8; i ++ ) {
        if( grid[i][y] == 0 ) {
            addQueen( i, y, true );
            //add queen and set all threatened elements of grid[][] to 1
            if( y == 7 ) {
                return true;
                // we have placed all queens successfully
            } else {
                if( solve( y + 1 ) ) {
                    return true; // we solved it,
                } else {
                    addQueen( i, y, false ); // remove the queen
                }
            }
        }
    }
    return false;
}
```

-- *page intentionally empty* --

3. String Matching (20 points)

The following is a shift table for the Knuth-Morris-Pratt algorithm. Use it to find the pattern in the given text. Show each position that the pattern is in.

		a	n	a	n	a	s
index	-1	0	1	2	3	4	5
shift	1	1	2	2	2	2	6

Text: banana picker

4. Graphs (30 points)

The following is the adjacency matrix representation of a graph. The graph is undirected, and hence the bottom half of the matrix is not given.

- Draw the graph as a diagram (circles as nodes, lines as edges).
- Perform a depth-first search on it, starting at node 1. If a node is connected to more than one other node, always access the one with the lowest index first.
- Write an algorithm that performs a depth-first search on a graph in matrix representation and prints out the nodes as it accesses them. The algorithm is not allowed to convert the graph to another notation. Hint: be sure to avoid an infinite loop in your algorithm.

	1	2	3	4	5	6	7	8
1		1	1	1	0	0	0	1
2			1	0	1	1	0	0
3				0	0	0	0	0
4					0	0	1	1
5						1	0	0
6							0	0
7								1
8								

--page intentionally empty --

5. Dynamic Programming (20 points extra credit)

Write an algorithm that calculates the n-th Fibonacci number by using recursion with a memory function.

The definition of the Fibonacci function is: $f(1) = f(2) = 1$; $f(n) = f(n-1) + f(n-2)$.