Name: _____

Student Number: _____

COMP SCI 2ME3 (Software Design I) Michael Soltys

DAY CLASS
DURATION OF EXAMINATION: 2 Hours
MCMASTER UNIVERSITY FINAL EXAMINATION April 2010

THIS EXAMINATION PAPER CONSISTS OF 3 PAGES AND 6 QUESTIONS.

YOU ARE RESPONSIBLE FOR ENSURING THAT YOUR COPY OF THE PAPER IS COMPLETE.
BRING ANY DISCREPANCY TO THE ATTENTION OF YOUR INVIGILATOR.

No aids allowed.

All questions worth 10 marks, for a total of 60.

Q1. Consider algorithm 1, where the assignments in lines 1 and 8 and done left to right.

Algorithm 1 Euclid's extended algorithm.

Pre-condition: $m > 0, n > 0$

1: $a \leftarrow 0$; $x \leftarrow 1$; $b \leftarrow 1$; $y \leftarrow 0$; $c \leftarrow m$; $d \leftarrow n$
2: **loop**
3: $q \leftarrow \mathrm{div}(c, d)$
4: $r \leftarrow \mathrm{rem}(c, d)$
5: **if** $r = 0$ **then**
6: stop
7: **end if**
8: $c \leftarrow d$; $d \leftarrow r$; $t \leftarrow x$; $x \leftarrow a$; $a \leftarrow t - qa$; $t \leftarrow y$; $y \leftarrow b$; $b \leftarrow t - qb$
9: **end loop**

Post-condition: $am + bn = d = \gcd(m, n)$

Show *partial* the correctness of algorithm 1. **Hint:** start by proving the following loop invariant:

$$am + bn = d, \quad xm + yn = c, \quad d > 0, \quad \gcd(c, d) = \gcd(m, n).$$

Q2. Suppose that we are given a graph $G = (V, E)$, a designated start node s, and a cost function for each edge $e \in E$, denoted $c(e)$. We are asked to compute the cheapest paths from s to every other node in G, where the cost of a path is the sum of the costs of its edges.

Consider the following greedy algorithm: the algorithm maintains a set S of explored nodes, and for each $u \in S$ it stores a value $d(u)$, which is the cheapest path inside S, starting at s and ending at u. Initially, $S = \{s\}$ and $d(s) = 0$. Now, for each $v \in V - S$ we find the shortest path to v by traveling inside the explored part S to some $u \in S$, followed by a single edge (u, v). See figure 1.

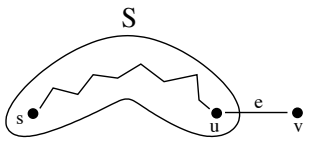

Figure 1: Computing the shortest path.

That is, we compute $d'(v) = \min_{u \in S, e=(u,v)} d(u) + c(e)$. We choose the node $v \in V - S$ for which this quantity is minimized, add v to S, and set $d(v) = d'(v)$, and repeat. Thus we add one node at a time to the explored part, and we stop when $S = V$.

Show, using the notion of a promising set, that this greedy algorithm computes the value of the shortest path $d(u)$ from s to every node u.

Q3. Consider the following divide and conquer version of the extended Euclid's algorithm:

Algorithm 2 `EUCLID`

Pre-condition: $m > 0, n \geq 0$

1: $a \leftarrow m$; $b \leftarrow n$
2: **if** $b = 0$ **then**
3: **return** $(a, 1, 0)$
4: **else**
5: $(d, x, y) \leftarrow$ `EUCLID`$(b, \text{rem}(a, b))$
6: **return** $(d, y, x - \text{div}(a, b) \cdot y)$
7: **end if**

Post-condition: $ax + by = d = \gcd(m, n)$

Show that this algorithm is *correct*. Note that each stage of the recursion (in particular the last one) returns three values; this is what we mean in line 5. Thus, in line 5 we get the appropriate values for b and $\text{rem}(a, b)$ and in line 6 we return those values modified to serve for a and b.

Q4. Consider the algorithm for the simple knapsack problem (SKS).

Algorithm 3 Simple knapsack (SKS)

```
1: S(0) ⟵ T
2: for j : 1..C do
3:         S(j) ⟵ F
4: end for
5: for i : 1..d do
6:         for decreasing j : C..1 do
7:                 if (j ≥ w_i and S(j − w_i) = T) then
8:                         S(j) ⟵ T
9:                 end if
10:        end for
11: end for
```

(a) Recall that $R(i, j) = \mathsf{T}$ iff the total weight j can be constructed with a *subset* of the first i weights. Give the recurrence for computing R.

(b) The assertion $S(j) = R(i, j)$ can be proved by induction on the number of times the i-loop in algorithm 3 is executed. This assertion implies that upon termination of the algorithm, $S(j) = R(d, j)$ for all j. Prove this formally, by giving pre/post-conditions, a loop invariant, and a standard proof of correctness.

Q5. Show that any page replacement algorithm, whether online or offline, can be modified to be demand paging without increasing the overall cost of any request sequence.

Q6. Explain the Diffie-Hellman key exchange protocol. Show that this protocol can be broken if we have an efficient way of solving the discrete log problem.

END OF EXAM