NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

SEMESTER II: 2008–2009

EXAMINATION FOR

GEM 1501 – Problem Solving for Computing

Tuesday 28 April 2009 Morning – Time Allowed 2 Hours

INSTRUCTIONS TO CANDIDATES

1. This examination paper consists of TEN (10) questions and comprises ELEVEN (11) printed pages.
2. Answer **ALL** questions.
3. This is a **Closed Book** examination.
4. Every question counts FIVE (5) marks which are distributed equally on sub-questions in the case that there are any. The maximum possible marks are 50.
5. Please write your Matriculation Number below:

   MATRICULATION NO: ______________________________

This portion is for examiner's use only

Qestion	Marks	Remarks	Qestion	Marks	Remarks
Q01:			Q06:		
Q02:			Q07:		
Q03:			Q08:		
Q04:			Q09:		
Q05:			Q10:		
			Total:		

Question 1 [5 marks] **GEM 1501**

In what time did the following scientists and engineers make their contributions to algorithmics and computing?

(a) When did Charles Babbage try to construct his Analytical Engine?

☐ Before 500 ☐ 500-1500 ☐ 1501-1800 ☒ 1801-1900
☐ 1901-1930 ☐ 1931-1960 ☐ 1961-1990 ☐ After 1990

(b) When did Euclid develop his famous algorithm?

☒ Before 500 ☐ 500-1500 ☐ 1501-1800 ☐ 1801-1900
☐ 1901-1930 ☐ 1931-1960 ☐ 1961-1990 ☐ After 1990

(c) When did Hermann Hollerith invent his tabulating machine for the American Census?

☐ Before 500 ☐ 500-1500 ☐ 1501-1800 ☒ 1801-1900
☐ 1901-1930 ☐ 1931-1960 ☐ 1961-1990 ☐ After 1990

(d) When did Mohammed al-Khowârizmî formulate his algorithms for algebraic problems like solving quadratic equations?

☐ Before 500 ☒ 500-1500 ☐ 1501-1800 ☐ 1801-1900
☐ 1901-1930 ☐ 1931-1960 ☐ 1961-1990 ☐ After 1990

(e) Fortran is the oldest programming language which is still in use. When was its first version developed?

☐ Before 500 ☐ 500-1500 ☐ 1501-1800 ☐ 1801-1900
☐ 1901-1930 ☒ 1931-1960 ☐ 1961-1990 ☐ After 1990

Question 2 [5 marks] **GEM 1501**

Categorize the following problems in (a), (b), (c), (d) and (e) below as P, NP-complete, PSPACE-complete and EXPTIME-complete. Here “P” means “known to be in P”.

(a) The problem 2SAT of all formulas which are conjunctions of clauses with up to 2 literals.

[x] P [ ] NP-complete [ ] PSPACE-complete [ ] EXPTIME-complete.

(b) The problem 5SAT of all formulas which are conjunctions of clauses with up to 5 literals.

[ ] P [x] NP-complete [ ] PSPACE-complete [ ] EXPTIME-complete.

(c) A winning strategy for the game checkers on an $n \times n$ board.

[ ] P [ ] NP-complete [ ] PSPACE-complete [x] EXPTIME-complete.

(d) The travelling salesman problem where it is asked whether there is a round tour through given cities of a length below a given bound.

[ ] P [x] NP-complete [ ] PSPACE-complete [ ] EXPTIME-complete.

(e) The problem whether a given number in decimal notation is prime.

[x] P [ ] NP-complete [ ] PSPACE-complete [ ] EXPTIME-complete.

Question 3 [5 marks] **GEM 1501**

Which of the following sets are decidable, recursively enumerable but undecidable and not recursively enumerable at all?

(a) The set of all Java Script programs which do not contain any while-loop.
[x] decidable [ ] r.e. and undecideable [ ] not r.e.

(b) The set of all P such that P is a Java Script program for which $P(P(88))$ halts, that is, P on input 88 halts and produces an output x and P on input x halts as well and produces an output y.
[ ] decidable [x] r.e. and undecideable [ ] not r.e.

(c) The set of all Java Script programs which halt at least on one input.
[ ] decidable [x] r.e. and undecideable [ ] not r.e.

(d) The set of all Java Script programs which halt on all even inputs.
[ ] decidable [ ] r.e. and undecideable [x] not r.e.

(e) The set of all Boolean formulas which have 5 satisfying assignments.
[x] decidable [ ] r.e. and undecideable [ ] not r.e.

Determine the order of the following expressions; take the optimal choice, so tick $O(n^2)$ for $3n^2$ and not $O(n^3)$.

(a) $n\log^2(n) + n^2\log(n) + n^3$.

[ ] $O(\log(n))$ [ ] $O(\log^2(n))$ [ ] $O(\log^3(n))$ [ ] $O(n)$ [ ] $O(n\log(n))$
[ ] $O(n\log^2(n))$ [ ] $O(n^2)$ [ ] $O(n^2\log(n))$ [ ] $O(n^2\log^2(n))$ [x] $O(n^3)$
[ ] $O(2^n)$ [ ] $O(3^n)$ [ ] $O(4^n)$ [ ] $O(5^n)$ [ ] $O(n^n)$

(b) $(n+1)\cdot(n+2)\cdot(n+3)$.

[ ] $O(\log(n))$ [ ] $O(\log^2(n))$ [ ] $O(\log^3(n))$ [ ] $O(n)$ [ ] $O(n\log(n))$
[ ] $O(n\log^2(n))$ [ ] $O(n^2)$ [ ] $O(n^2\log(n))$ [ ] $O(n^2\log^2(n))$ [x] $O(n^3)$
[ ] $O(2^n)$ [ ] $O(3^n)$ [ ] $O(4^n)$ [ ] $O(5^n)$ [ ] $O(n^n)$

(c) $(n+\log(n))\cdot(n+\log(n))\cdot(\log(n)+\log^2(n))$.

[ ] $O(\log(n))$ [ ] $O(\log^2(n))$ [ ] $O(\log^3(n))$ [ ] $O(n)$ [ ] $O(n\log(n))$
[ ] $O(n\log^2(n))$ [ ] $O(n^2)$ [ ] $O(n^2\log(n))$ [x] $O(n^2\log^2(n))$ [ ] $O(n^3)$
[ ] $O(2^n)$ [ ] $O(3^n)$ [ ] $O(4^n)$ [ ] $O(5^n)$ [ ] $O(n^n)$

(d) $2^{n+123456789} + 3^{n+1234321}$.

[ ] $O(\log(n))$ [ ] $O(\log^2(n))$ [ ] $O(\log^3(n))$ [ ] $O(n)$ [ ] $O(n\log(n))$
[ ] $O(n\log^2(n))$ [ ] $O(n^2)$ [ ] $O(n^2\log(n))$ [ ] $O(n^2\log^2(n))$ [ ] $O(n^3)$
[ ] $O(2^n)$ [x] $O(3^n)$ [ ] $O(4^n)$ [ ] $O(5^n)$ [ ] $O(n^n)$

(e) $5\log(n) + 4\log^2(n) + 3\log^3(n) + 2n + 1n\log(n) + 0n\log^2(n)$.

[ ] $O(\log(n))$ [ ] $O(\log^2(n))$ [ ] $O(\log^3(n))$ [ ] $O(n)$ [x] $O(n\log(n))$
[ ] $O(n\log^2(n))$ [ ] $O(n^2)$ [ ] $O(n^2\log(n))$ [ ] $O(n^2\log^2(n))$ [ ] $O(n^3)$
[ ] $O(2^n)$ [ ] $O(3^n)$ [ ] $O(4^n)$ [ ] $O(5^n)$ [ ] $O(n^n)$

Question 5 [5 marks] **GEM 1501**

What is the status of the following mathematical statements? The answer can be YES if scientists know that the statement is true, the answer can be NO if scientists know that the answer is false and the answer can be OPEN if scientists do not know the answer and it is an important open problem.

(a) Is P = PSPACE?
☐ YES, ☐ NO, ☒ OPEN.

(b) Is there an undecidable set A such that both A and its complement are recursively enumerable?
☐ YES, ☒ NO, ☐ OPEN.

(c) Are there decidable problems which are not in EXPTIME?
☒ YES, ☐ NO, ☐ OPEN.

(d) Can every problem in P be accepted by a finite automaton?
☐ YES, ☒ NO, ☐ OPEN.

(e) Is Nick's class NC different from EXPTIME?
☒ YES, ☐ NO, ☐ OPEN.

Question 6 [5 marks] **GEM 1501**

Run resolution on the following set of clauses:
$x_1 \vee x_4$, $x_1 \vee x_5$, $x_2 \vee x_3$, $x_2 \vee x_4$, $x_3 \vee x_4$, $\neg x_4$, $x_5 \vee x_6$, $\neg x_5 \vee x_6$, $\neg x_6$.

(–) [x] Set x_1 true [ ] Set x_1 false [ ] Resolve x_1.

New clauses: $x_2 \vee x_3$, $x_2 \vee x_4$, $x_3 \vee x_4$, $\neg x_4$, $x_5 \vee x_6$, $\neg x_5 \vee x_6$, $\neg x_6$.

(a) [x] Set x_2 true [ ] Set x_2 false [ ] Resolve x_2.

$x_3 \vee x_4$, $\neg x_4$, $x_5 \vee x_6$, $\neg x_5 \vee x_6$, $\neg x_6$.

(b) [x] Set x_3 true [ ] Set x_3 false [ ] Resolve x_3.

$\neg x_4$,$x_5 \vee x_6$, $\neg x_5 \vee x_6$, $\neg x_6$.

(c) [ ] Set x_4 true [x] Set x_4 false [ ] Resolve x_4.

$x_5 \vee x_6$, $\neg x_5 \vee x_6$, $\neg x_6$.

(d) [ ] Set x_5 true [ ] Set x_5 false [x] Resolve x_5.

x_6, $\neg x_6$.

(e) So the formula is [ ] satisfiable [x] unsatisfiable.

Five programmers submitted the following programs for computing the Fibonacci numbers. The input n is a natural number. Evaluate the proposed programs as "Okay", "Exponential time" (in the parameter n, not in size of n), "Has syntax-errors" and "Not terminating". A program which needs exponential time is not okay as it can be done in polynomial time.

(a)
```
function fibonaccia(n)
  { return(fibonaccia(n-1)+fibonaccia(n-2)); }
```

☐ Okay; ☐ Exponential time; ☐ Has syntax-errors; ☒ Not terminating.

(b)
```
function fibonaccib(n)
  { var ar = new Array(0,1,1,2);
    while (ar.length <= n)
      { ar.push(ar[ar.length-1]+ar[ar.length-2]); }
    return(ar[n]); }
```

☒ Okay; ☐ Exponential time; ☐ Has syntax-errors; ☐ Not terminating.

(c)
```
function fibonaccic(n)
  { var m=0; var k=1; var h; var o=n;
    while (o>0)
      { h=m+k; m=k; k=h; }
    return(m); }
```

☐ Okay; ☐ Exponential time; ☐ Has syntax-errors; ☒ Not terminating.

(d)
```
function fibonaccid(n)
  { var m=0; var k=1; var h; var o;
    for (o=0;o<n)
      { h=m+k; m=k; k=h; }
    return(m); }
```

☐ Okay; ☐ Exponential time; ☒ Has syntax-errors; ☐ Not terminating.

(e)
```
function fibonaccie(n)
  { if (n<1) { return(0); }
    if (n<3) { return(1); }
    return(fibonaccie(n-3)+2*fibonaccie(n-2)); }
```

☐ Okay; ☒ Exponential time; ☐ Has syntax-errors; ☐ Not terminating.

Question 8 [5 marks] **GEM 1501**

Analyze the following program which accepts or rejects words (by returning the value `"accept"` or `"reject"`, respectively). The input is always a string (that is, text).

```
function wordcheck(x)
  { var count = 0; var n=x.length; var m;
    for (m=0;m<n;m++)
      { if (x.charAt(m)=="(") { count++; }
        if (x.charAt(m)==")") { count--; }
        if (count<0) { return("reject"); } }
    if (count>0) { return("reject"); }
    else { return("accept"); } }
```

What can be said about the program.

(a) On input `"(aa(bb)+cc(bb))"`

- [x] the program outputs `"accept"`;
- [ ] the program outputs `"reject"`;
- [ ] the program does not terminate.

(b) On input `"(aa(bb)+cc(bb))))"`

- [ ] the program outputs `"accept"`;
- [x] the program outputs `"reject"`;
- [ ] the program does not terminate.

(c) Let L be the set of all strings on which the program `"wordcheck"` terminates and outputs `"accept"`.

- [x] L contains all balanced expressions of brackets;
- [ ] L contains all expressions with more opening than closing brackets;
- [ ] L contains all expressions with as many opening as closing brackets.

(d) Which statement on the set L is true:

- [ ] L can be accepted by a finite automaton;
- [x] L can be accepted by a one-stack machine but not by a finite automaton;
- [ ] L cannot be accepted by a one-stack machine.

(e) Evaluate the order of the runtime of the program where basic JavaScript commands like adding or accessing members of a string count as 1 time unit. The order of the runtime is

[ ] $O(\log(n))$ [x] $O(n)$ [ ] $O(n \log(n))$ [ ] $O(n^2)$ [ ] $O(2^n)$

[ ] $O(\infty)$, that is, the program does sometimes not terminate.

Here n is the length of the input word `x`, that is, the value `x.length`.

Question 9 [5 marks] **GEM 1501**

Write a Java Script function which computes how many square numbers are between m and n. So count$(0, 4)$ is 3 as there are the square numbers 0, 1 and 4 between 0 and 4. Similarly count$(5, 8)$ is 0 and count$(4, 25)$ is 4. If $n < 0$ or $n < m$ then count(m, n) should be 0 as there cannot be any square number k^2 with $m \leq k^2 \leq n$.

```
function count(m,n)
  { var c;
    var k; c=0;
    for (k=0;k<=n;k++)
      { if ((m<=k*k)&&(k*k<=n))
          { c++; } }
    return(c); }
```

Question 10 [5 marks] **GEM 1501**

Assume that an array ar of length n is given. Write a Java Script function which checks how many numbers appear in the array exactly 2 times. So if ar equals $(1, 2, 1, 2, 9, 3, 2, 9, 5, 8, 8)$ then the return value of the function is 3 as $1, 8, 9$ appear in the array exactly twice.

```
function pairnum(ar)
  { var n = ar.length; var m = 0;
    var i; var j; var k; var h;
      { for (i=0;i<n;i++)
          { for (j=i+1;j<n;j++)
              { if (ar[i]==ar[j])
                  { h=1;
                    for (k=0;k<n;k++)
                      { if ((k!=i)&&(k!=j)&&(ar[k]==ar[i]))
                          { h=0; } }
                    m += h; } } } }
    return(m); }
```

END OF PAPER