NATIONAL UNIVERSITY OF SINGAPORE

SCHOOL OF COMPUTING

SEMESTER II: 2005–2006

MIDTERM EXAMINATION FOR

GEM 1501 – Problem Solving for Computing

Friday 24 March 2006 Morning – Time Allowed 45 Minutes

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 an **Closed Book** examination.
4. Every question counts TWELVE (12) Marks which is distributed equally on subquestions in the case that there are any. The maximum possible marks are 120.
5. Please write your Matriculation Number below:

MATRICULATION NO: _____

This portion is for examiner's use only

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

Question 1 [12 marks] **GEM 1501**

Look at the following sample Java Script program.

```
var n;
do { n = window.prompt("Input a number",17+4);
     window.alert("The square of "+n+" is "+(n*n)+"."); }
   while(window.confirm("Do you want to go on?"));
```

Describe what the program is doing and what the three commands starting with "window" are used for:

- Overall description of program behaviour: The program computes the squares of inputs given by the user; after each round it asks the user whether he wants to go on.
- window.prompt: Reads an input from the keyboard; in the case that the user just presses return, a default number is taken which is $17 + 4$, that is, 21, in this program.
- window.alert: It informs the user about some information, in this program the information is the square of the number n just read by the user.
- window.confirm: In this option the user can either click "OK" or "Cancel"; the program above runs as long as the user clicks "OK".

What has the user to do in order to terminate the program? The user has to click "Cancel" at the box opened by the command "window.confirm".

What is the value of the variable n if the user just hits the return key after seeing the first window generated by "window.prompt"? __21__ .

Question 2 [12 marks] **GEM 1501**

In a club, they keep a record on all members of the club, including those who deceased or quitted. On becoming a member, everyone receives a member number which consists of the order in which he enters plus three decimal random digits added, for example the record of the first member of the club is `("Kelvin","Edward",273)` and the record of the seventeenth member of the club is `("Lawrence","Anna",16734)`. These additional digits are added in order to make it improbable that someone can access the club's data base by inventing a membership number. The records are all stored in an array `a`. If, for example, Anna Lawrence is the record number 5 in this array then `a[5].surname` is "Lawrence", `a[5].firstname` is "Anna" and `a[5].number` is 16734. Write a function which produces a sorted version `b` of this array in linear time such that the member numbers are in ascending order. The method "`a.sort`" of JavaScript should not be used.

```
function sortedcopy(a)
  { var k;

    ------

       var b = new Array(a.length);
       for (k=0;k<a.length;k++)
         { b[Math.floor(a[k].number*0.001)] = a[k]; }

    -----

    return(b); }
```

Question 3 [12 marks] **GEM 1501**

The following functions e, f, g and h sum up some elements of a three-dimensional array `a`. How much time do the functions e, f, g, h use? This should be given in terms of the order in dependence of n, for example Merge Sort uses $O(n \cdot \log(n))$.

```
function e(n)
  { var i; var j; var s=0;
    for (i=0;i<n;i++)
      { for (j=0;j<n;j++)
          { s += a[i][j][0]; } }
    return(s); }
```

Give the order of the time-usage of the function e depending on n: $O(n^2)$.

```
function f(n)
  { var i; var j; var k; var s=0;
    for (i=0;i<n;i++)
      { for (j=0;j<n;j++)
          { for (k=1;k<n;k=k+k)
              { s += a[i][j][k]; } } }
    return(s); }
```

Give the order of the time-usage of the function f depending on n: $O(n^2 \log(n))$.

```
function g(n)
  { var k = n*n*n; var s=0;
    while (k>0)
      { s+= a[k][0][0]; k--; }
    return(s); }
```

Give the order of the time-usage of the function g depending on n: $O(n^3)$.

```
function h(n)
  { var k; var s = f(n*n*n)+g(n*n*n);
    for (k=0;k<=n;k++)
      { s += f(k)*g(n-k)+f(n-k)*g(k); }
    return(s); }
```

Give the order of the time-usage of the function h depending on n: $O(n^9)$.

Question 4 [12 marks] **GEM 1501**

An example for an NP-complete problem are fixed-size monkey puzzles. Solve the following monkey puzzle and write down the conditions which must be satisfied by the solution (each correct tile one mark, each correct condition one mark):

```
26 82 67
71 77 89

48 77 71
67 91 19

89 91 19
45 55 53
```

What are the conditions a solution has to satisfy?
The size of the task and solution are the same: [x] Yes, [ ] No.
It is permitted to use some tiles more often and others less often in order to get a solution: [ ] Yes, [x] No.
What conditions have to be satisfied for any pair of neighbouring tiles? Write it down in terms of conditions on a,b,c,d,e,f,g,h, the condition is in both cases (next to each other and above each other) the same: $b = e$ and $d = g$.

```
Case "Next To Each Other"
 ab ef
 cd gh
```

```
Case "Above Each Other"
 ac
 bd

 eg
 fh
```

Question 5 [12 marks] **GEM 1501**

Run the resolution algorithm processing the variables x_1, x_2, x_3, x_4, x_5 in this order for the following two examples. Recall that processing x_i means to modify the set of clauses according to the resolution algorithm in a way such that the variable x_i does no longer occur in the set of clauses and that the previous set is satisfiable iff the new set is satisfiable. Write the resulting formulas after each step and use "false" for the unsatisfiable empty clause and "$\emptyset$" for an empty set of clauses — these conventions are needed after processing all variables and occur at opposite outcomes of the algorithm.

First Set of Clauses:

- $x_1 \vee x_2$; $\neg x_1 \vee x_3$; $\neg x_2 \vee x_3$; $x_3 \vee x_4$; $\neg x_4 \vee x_5$; $\neg x_5$
- After processing x_1:
- $x_2 \vee x_3$; $\neg x_2 \vee x_3$; $x_3 \vee x_4$; $\neg x_4 \vee x_5$; $\neg x_5$
- After processing x_2:
- x_3; $x_3 \vee x_4$; $\neg x_4 \vee x_5$; $\neg x_5$
- After processing x_3: $\neg x_4 \vee x_5$; $\neg x_5$
- After processing x_4: $\neg x_5$
- After processing x_5: no clauses left
- The given instance is satisfiable: [x] Yes, [ ] No.

Second Set of Clauses:

- $x_1 \vee x_2$; $x_1 \vee x_3$; $\neg x_2$; $x_3 \vee x_5$; $\neg x_3 \vee x_4 \vee x_5$; $\neg x_3 \vee \neg x_4$; $\neg x_5$
- After processing x_1: $x_2 \vee x_3$; $\neg x_2$; $x_3 \vee x_5$; $\neg x_3 \vee x_4 \vee x_5$; $\neg x_3 \vee \neg x_4$; $\neg x_5$
- After processing x_2: x_3; $x_3 \vee x_5$; $\neg x_3 \vee x_4 \vee x_5$; $\neg x_3 \vee \neg x_4$; $\neg x_5$
- After processing x_3: $x_4 \vee x_5$; $\neg x_4$; $\neg x_5$
- After processing x_4: x_5; $\neg x_5$
- After processing x_5: empty clause (false)
- The given instance is satisfiable: [ ] Yes, [x] No.

Question 6 [12 marks] **GEM 1501**

Determine the degree of decidability or undecidability of the following sets of texts T. Check "incomplete r.e." only if the corresponding set is undecidable, for decidable sets check "decidable".

The set $\{T : T$ is a syntactically correct text of a Java Script program$\}$ is:
☐ not r.e., ☐ complete r.e., ☐ incomplete r.e., ☒ decidable.

The set $\{T : T$ is a compressible text$\}$ is:
☐ not r.e., ☐ complete r.e., ☒ incomplete r.e., ☐ decidable.

The set $\{T : T$ is a syntactically correct text of a Java Script program which never asks for any input and which terminates$\}$ is:
☐ not r.e., ☒ complete r.e., ☐ incomplete r.e., ☐ decidable.

The set $\{T : T$ is a syntactically correct text of a Java Script program which asks for some input and which terminates for any input$\}$ is:
☒ not r.e., ☐ complete r.e., ☐ incomplete r.e., ☐ decidable.

The set $\{T : T$ is a text which consists of a list of some clauses forming an unsatisfiable Boolean formula$\}$ is:
☐ not r.e., ☐ complete r.e., ☐ incomplete r.e., ☒ decidable.

The set $\{T : T$ is a syntactically correct text of a Java Script program which asks for some input and terminates if the input is 0 or 1 but not on other inputs$\}$ is:
☒ not r.e., ☐ complete r.e., ☐ incomplete r.e., ☐ decidable.

Question 7 [12 marks] **GEM 1501**

Assume that you are permitted only to use Java Script functions and while-loops, addition, subtraction, comparisons and integer constants. Complete the below functions such that they do what is indicated in the comments. Do not use more space than provided and do not write informally things like "add s 125 times". If the space does not suffice, then there is a better algorithm.

```
// computing the product of m and n
function prod(m,n)
  { var k = _____m____; var s = 0;
    while (k>0)
      { s = s + _____n______; k = k--; }
    return(___s____); }

// computing 125 to the power of n, n is a natural number
// power125(0) is 1, power125(1) is 125, power125(2) is 15625
function power125(n)
  { var m = ____n+n+n______; var s = 1;
    while (m>0)
      { s = s + ____________s+s+s+s_____________; m--; }
    return(____s___); }

// computing the down-rounded root of n, n is a natural number
function root(n)
  { var m = _____0______; var s = 0;
    while (s<=n)
      { s = s + _______m+m+1________; m++; }
    return(___m-1____); }

// computing the down-rounded binary logarithm of n,
// n is a natural number, n>0; so logarithm is at least 0
function log(n)
  { var m = ____0_____; var s = 2;
    while (s<=n)
      { s = s + ________s____________; m++; }
    return(____m____); }
```

Question 8 [12 marks] **GEM 1501**

Assume that a deterministic PDA (Push-Down Automaton or One-Stack Machine) accepts a word iff the stack after processing the word is empty, that the PDA has c finite states and that it accepts all palindromes. Write a proof that the given PDA also accepts some words which are not a palindrome.

Proof: Recall that a word is a palindrome iff it is the same when read forward or read backward. Furthermore, recall that a^n just means a sequence of n times the letter a, so a^5 is $aaaaa$.

The words a, aa, aaa are palindromes and so is every word a^n. Furthermore aba, $aabaa$ and, in general, a^nba^n are palindromes as well, but a^5ba^7 is not a palindrome.

As the PDA accepts these words, the stack is empty after reading a^n for every n. Now consider the palindrome a^nba^n for some $n > c$. The PDA accepts this palindrome as well. As $n > c$ there are two i, j such that $0 \leq i < j \leq n$ with the PDA being in the same state after reading a^i and a^j. As the stack is in both cases empty, the PDA does the same after having seen i and having seen j times an a. Therefore the PDA also accepts the word $a^ia^{n-j}ba^n = a^{n-(j-i)}ba^n$ which is not a palindrome as $n-(j-i) < n$.

Question 9 [12 marks] **GEM 1501**

Let 3CLIQUE be the following problem: For given size n, the input is a $n \times n$ matrix A with entries 0 and 1. Now A is a 3CLIQUE instance if there are 3 rows i, j, k with $i < j < k$ and $A[i][j]$, $A[i][k]$, $A[j][k]$ all being a 1. Prove that 3CLIQUE is in NC by giving an outline of a network (for given length n) which checks whether an $A * A$ matrix at the input is a 3CLIQUE instance or not.

Gates can have 1, 2 or 3 inputs and can compute an "and", "or" or "not". The input layer has n^2 gates with gate (i, j) being 1 iff $A[i][j]$ is 1 and $i < j$; gate (i, j) is 0 otherwise. The output gate should have 1 iff the matrix at the input is a 3CLIQUE instance.

For keeping things simple, assume that n is a power of 2, that is, that n equals to 2^m for some m. Please state besides the network also the number of layers and gates in dependence of n and m: $3m+1$ layers and $2^{3m+1}-1$ gates. If you do not know how to make the network but can at least give a reasonable estimates which are neither too small nor too large, you get 4 marks.

Solution: In the first layer there are n^3 gates, for each triple (i, j, k) with $0 \leq i < j < k < n$ one such that the gate $G_{i,j,k}$ which takes the "AND" over $A[i][j]$, $A[i][k]$ and $A[j][k]$. For each triple (i, j, k) with $i, j, k < n$ but not $i < j < k$ let $G_{i,j,k}$ be always 0. Now there are m^3 layers, each of them having half of the gates of the layer below such that each two gates of the layer below are combined by one gate taking the "OR" of these two gates in the next layer. So there are in total $3m+1$ layers and the layer number $k = 0, 1, 2, \ldots, 3m$ has 2^{3m-k} gates. This gives $2^{3m+1} - 1$ gates in total.

Question 10 [12 marks] **GEM 1501**

Terminology of Coordinating parallel processes.

How it is called if a process never receives computing resources because others have higher priority?

☐ Concurrency ☐ Conflict ☐ Deadlock ☒ Starvation

How it is called if several processes run at the same time?

☒ Concurrency ☐ Conflict ☐ Deadlock ☐ Starvation

How it is called if two processes do wait with their start for the other one to start and terminate first such that at the end none of these two processes does anything?

☐ Concurrency ☐ Conflict ☒ Deadlock ☐ Starvation

How it is called if two processes want to update the same variable at the same time to two different values?

☐ Concurrency ☒ Conflict ☐ Deadlock ☐ Starvation

How are variables called which are used in protocols for assigning resources to computing processes? Semaphores .

Furthermore, are such variables always Boolean or can they also be integer variables?

☐ Always Boolean; ☒ Sometimes also integers.

END OF PAPER