Midterm Examination 2
GEM 1501: Problem Solving for Computing

08.04.2009, 12.00-12.30h

Matriculation Number: _____

Rules
Each correct question, 1 mark. Maximum score: 12 marks.
Programming Language for Questions 8–12 is Java Script.

Question 1. What is the product complexity of an algorithm?

☐ The product of the length of the computer program and its runtime on data of length n;

☒ The product of the number of processors used in parallel computation and the parallel time of the algorithm;

☐ The product of time and space used by the algorithm.

Question 2. Use the Theorem of Rice in order to define a set A which is undecidable and complete the definition below:

$A = \{P : P(0) \text{ halts and takes the value } 88\}$

$A = \{P : P \text{ halts on all inputs}\}$

$A = \{P : P \text{ does not halt on infinitely many inputs}\}$

$A = \{P : \text{the set of } x \text{ on which } P \text{ halts is undecidable}\}$

Note that A should not only be undecidable but also satisfy the requirement of Rice's Theorem, that is, two equivalent programs P, Q should either be both in A or both outside A.

The last solution seems to have been the favourate one of students of this course. One of these solutions is sufficient.

Question 3. Which of the following chains of inclusions is known to be correct?

☐ $P = RP = NP \subset EXP$;

☒ $P \subseteq RP \subseteq NP \subseteq EXP$;

☐ $P \subseteq NP \subseteq RP \subseteq EXP$;

☐ $P \subset NP = RP = EXP$.

Question 4. Complete the following definition of Nick's Class (NC).

A problem A is in NC if and only if there is a polynomial time algorithm F which produces for every n in time polynomially in n a network $F(n)$ solving all instances of length n such that $F(n)$ has

☐ logarithmically ☐ polylogarithmically
☒ polynomially ☐ exponentially

many gates arranged in

☐ logarithmically ☒ polylogarithmically
☐ polynomially ☐ exponentially

many layers where the inputs of the network are in the lowest layer and every gate has two inputs from

☒ layers below ☐ layers above ☐ layers below and above.

Question 5. Consider the finite automaton with starting state q_0, rejecting states q_0, q_2, accepting state q_1 and the following transition table:

State	Type	Successor at 0	Successor at 1	Successor at 2	Successor at 3
q_0	reject	q_0	q_1	q_0	q_0
q_1	accept	q_1	q_1	q_2	q_1
q_2	reject	q_2	q_2	q_1	q_2

Which of the following words are accepted by this finite automaton? Tick 2 out of 5.

☐ 0000 ☒ 0101 ☐ 0202 ☒ 1223 ☐ 0123.

Question 6. Let REG be the class of problems accepted by finite automata and LOGSPACE be the class of problems solvable by a three-tape Turing machine using logarithmic space. What is the relation between these two classes?

☐ REG = LOGSPACE; ☒ REG $\subset$ LOGSPACE ☐ REG $\supset$ LOGSPACE
☐ REG $\not\subseteq$ LOGSPACE and LOGSPACE $\not\subseteq$ REG.

Question 7. Which of the following statements are true about semaphores? Tick 3 out of 6 answers.

☒ Semaphores are used to coordinate parallel computing.
☐ Semaphores can generate random numbers.
☒ Semaphores are updated by small uninterruptible operations.
☒ Semaphores are variables which have always a well-defined value.
☐ Semaphores are variables which can only take numerical values.
☐ Semaphores are an important ingredient of public key cryptography.

Question 8. The command `c++` is used to increment the value of `c` by one and the increment is done after the variable had been used for other purposes; so if `c` is 5 and the statement `d=c++;` is done then `d` is 5 and `c` is 6 afterwards. The same command can also be done with more complicatedly organized variables. In the following, assume that `x` is `(1,2,3,4)` and `y` is `(4,5,6,7)`. What is the value of `x` after the statement `x[x[0]++]=y[x[2]++]++;` had been done?

☐ `(1,7,3,4)` ☒ `(2,7,4,4)` ☐ `(2,8,4,4)` ☐ `(2,2,8,4)`

Question 9. Consider the following algorithm to sort an array of numbers by choosing randomly a position and exchanging the neighbours (if needed):

```
function randomsort(a)
  { var n = a.length;
    if (n<2) { return; }
    var m; var k; var b;
    for (m=0;m<200*n;m=m+1)
      { k = Math.floor((n-1)*Math.random());
        if (a[k] > a[k+1]) { b = a[k+1]; a[k+1] = a[k]; a[k] = b; } }
    return; }
```

What can be said about this program? Tick exactly one answer and take the explanation below each answer into account; n is the length of the array a to be sorted by the function.

☐ The algorithm never terminates.

The reason is that the termination of the for-loop only succeeds if each time in the function "Math.random()" the number 0 is drawn.

☐ The algorithm proves that randomized sorting is faster than mergesort.

The reason is that the algorithm works with $200*n$ comparisons for any type of input while mergesort would need $n\log(n)$. So the algorithm is one of the examples where randomized algorithms are faster than deterministic ones.

☐ It is random whether the algorithm succeeds.

Due to a advanced case-by-case analysis one can show that the probability to sort the input correctly is at least $1/n$ (for each input of length $n > 1$). So one would have to run the algorithm approximately $\log(n)$ times until it succeeds to sort the data and so the overall performance until correct output would be $O(n\log(n))$.

☒ On some inputs the algorithm never works.

If $n = 3000$ and the first thousand digits are 2, the next thousand digits are 1 and the last thousand digits are 0 then the destination of each 0 is more than 1000 positions away and more than $1000000 = 1000*n/3$ exchanges and comparisons are needed to sort the data. But the algorithm does at most $200*n$ exchanges and cannot succeed.

Question 10. Consider the following program.

```
function eval(w)
  { var a=0; var b=0; var c=0; var n=w.length; var m;
    for (m=0;m<n;m++)
      { switch(w.charAt(m))
          { case "0": a++; break;
            case "1": b++; break;
            case "2": c++; break;
            default: break; } }
    if ((a==b)&&(a==c))
      { return(a+1); }
    else { return(0); } }
```

What can be said about this program?

☐ It needs PSPACE as it counts three independent things and that cannot be done with sublinear space.

☐ It needs POLYLOGSPACE but is properly contained in PSPACE.

☒ It needs only LOGSPACE as the 5 variables used (besides the input) range from 0 to n and can be stored in space $O(\log(n))$.

Question 11. Complete the following program to reverse an array `x`. The output should be an array `y`. `x` and `y` are global variables, hence the function has no inputs. Fill in the missing names of the operations.

```
function reverse()

  { y = ______new______ Array();

    var z;

    while (x.______length______ > 0)

      { z = x.______pop______();

        y.______push______(z); }

    return; }
```

Question 12. Consider a type of counter programs which can only use addition, subtraction and comparisons but not multiplication, division and other advanced operations. Write a counter program which computes the function $x \mapsto 2^{(x^2+1)}$, you can use the usual Java Script syntax and do not have to rely on line numbers. While-loops and for-loops are permitted. Other than in usual Java Script, all variables are expected to take only natural numbers as values.

```
function f(x)
  { var y;
-------------
    var z = 1;
    var u;
    for (u=0;u<x;u++)
      { z = z+x; }
    y=1;
    for (u=0;u<z;u++)
      { y = y+y; }
-------------
   return(y); }
```

MARKDISTRIBUTION

42 participants (one of them 0 marks for not showing up)

Marks:	0	1	2	3	4	5	6	7	8	9	10	11	12
Number of Students:	1	1	0	3	3	4	12	6	7	5	0	0	0

Worksheet

Do not remove this sheet from the test.

You can use this sheet to do calculations, but you should write the answers into the space provided. Answers found here are not evaluated.