Midterm Examination 2
GEM 1501: Problem Solving for Computing

Wednesday 07.04.2010, duration half an hour

Matriculation Number: _______________

Rules
Each correct question, 1 mark. Maximum score: 12 marks.
Programming Language for Questions 7–12 is Java Script.

Question 1. Computations modulo a prime number are important in computer science. Show that you understand how to do this. Compute $f(9)$ modulo 11 where $f(x) = 220*(x-1)*x*(x+1)+5*(x-2)*(x+2)*x^{1819}+242*x^{23}+50*(x-8)*x+5*x+18$. That is, find that number y such that there is an integer z with $f(9) = 11 * z + y$. The number y is

☐ 0 ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5
☐ 6 ☒ 7 ☐ 8 ☐ 9 ☐ 10.

Question 2. Which of the following problems on multivariate quadratic polynomials $f(x_1, x_2, \ldots, x_n)$ and $g(x_1, x_2, \ldots, x_n)$ with integer coeffitients between $-n^n$ and $+n^n$ are known to be in RP? More precisely, for which of the three formulas below there is a randomised polynomial time algorithm that outputs with high probability "yes" if the formula with the given parameter polynomials f and g is true and always "no" if the formula with parameters f and g is false?

☐ $\exists x_1, x_2, \ldots, x_n[f(x_1, x_2, \ldots, x_n) = g(x_1, x_2, \ldots, x_n)]$;
☒ $\exists x_1, x_2, \ldots, x_n[f(x_1, x_2, \ldots, x_n) \neq g(x_1, x_2, \ldots, x_n)]$;
☐ $\exists x_1, x_2, \ldots, x_n[f(x_1, x_2, \ldots, x_n) * g(x_1, x_2, \ldots, x_n) = 0]$.

Question 3. What is the best possible complexity of parallel sorting of n inputs?

☐ $O(n \log(n))$ processors and $O(\log \log(n))$ parallel time;
☒ $O(n)$ processors and $O(\log(n))$ parallel time;
☐ $O(\sqrt{n})$ processors and $O(\sqrt{n})$ parallel time;
☐ $O(\log(n))$ processors and $O(n \log^3(n))$ parallel time.

Question 4. Which of the following problems are known to be solvable in NC? Check two out of five:

- [ ] Finding the shortest roundtour of n cities on a map;
- [ ] Computing greatest common divisor of two natural numbers;
- [x] Multiplication two $n * n$ matrices;
- [ ] Playing checkers on an $n * n$ board optimally;
- [x] Summation of n numbers.

Question 5. Consider the following complexity classes: LOGSPACE, NC, NP, P, PSPACE, RP. Find two of these classes which are known to be different:

The class ___LOGSPACE___ is different from the class ___PSPACE___.

Question 6. The set $\{T : T$ is the text of a program which halts on input 1965$\}$ is recursively enumerable but not decidable. Not every set is recursively enumerable. Please provide a definition of a set of texts which is not recursively enumerable.

The set $\{T : T$ is a text of a program which ___halts on all inputs___$\}$ is not recursively enumerable.

___Remark: There are many other possible solutions to this question.___

Question 7. Counter programs modify the value of a variable by at most one and all variables have natural numbers as a value. What is the function computed by the following counter program?

```
function f(n)
  { var m=2; var k=0; var s=0; var t;
    if ((n == 0)||(n == 1)) { return(n); }
    while (m != n) { m = m+1; k = k+1; }
    t = f(k); k = k+1; s = f(k);
    while(t != 0) { t = t-1; s = s+1; }
    return(s); }
```

The function value f(n) for input n is the following (check 1):

- [ ] the exponential function of n $(1, 2, 4, 8, 16, \ldots)$;
- [ ] the factorial of n, that is, $1 * 2 * \ldots * n$;
- [x] the n-th Fibonacci number $(0, 1, 1, 2, 3, 5, \ldots)$;
- [ ] the square of n;
- [ ] the sum $0 + 1 + \ldots + n$ of the numbers from 0 to n;
- [ ] the downrounded squareroot of n.

Question 8. Consider the following program where the input u is always a natural number:

```
function g(u)
  { var s = u%10;
    if (u > 9) { return(s+g(Math.floor(u*0.1+0.0005))); }
    else       { return(s); } }
```

Check the five correct answers:

- [ ] $g(12345)$ terminates and returns 10;
- [ ] $g(12345)$ terminates and returns 12;
- [x] $g(12345)$ terminates and returns 15;
- [ ] $g(12345)$ terminates and returns 19;
- [x] $g(u)$ is never strictly larger than u;
- [ ] $g(u)$ is equal to u for all u;
- [ ] $g(u)$ takes infinitely often the value 0;
- [x] $g(u)$ takes infinitely often the value 1;
- [x] $g(u)$ takes infinitely often the value 2;
- [x] $g(10 * u + r) = g(u) + r$ for all u and $r = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$;
- [ ] $g(10 * u + r) = g(u) * r$ for all u and $r = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$;
- [ ] $g(10 * u + r) = g(u)^r$ for all u and $r = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$.

Remark: the output is the sum of the decimal digits of the input.

Question 9. What is the runtime complexity of this program on inputs x,y where x+y has n decimal digits? Every input is a natural number.

```
function h(x,y)
  { if ((x<1)||(y<1)) { return(0); }
    var v = 0; var w = 0; var u = 0;
    while (u+y <= x)
      { while (u+v <= x) { w = v; v = v+v+y; }
        u = u+w; v = 0; w = 0; }
    return(x-u); }
```

The runtime complexity is

- [ ] Logarithmic or polylogarithmic in n;
- [x] Polynomial in n but not polylogarithmic in n;
- [ ] Exponential in n but not polynomial in n;
- [ ] Worse than exponential in n.

Question 10. Write a program which finds the largest prime number p below the input n; the program should return n if n is prime number. In the case that $n < 2$ the program should return 0.

```
function primesearch(n)
  { var p = n; var k=n-1;
    if (n<2) { return(0); }
    while ((k>1)&&(p>2))
      { if (p%k==0)
          { p--; k = p-1; }
        else {k--; } }
    return(p); }
```

Question 11. Write a program which finds the largest m such that in the array a some number occurs m times. So if a is $(1, 2, 3, 2, 2, 3, 4, 3, 3, 4, 53, 58)$ then $count(a)$ should return 4 as 3 occurs 4 times in a.

```
function count(a)
  { var n = a.length; var m; var i; var j; var k;
    m=0;
    for (i=0;i<n;i++)
      { k=0;
        for (j=0;j<n;j++)
          { if (a[i]==a[j]) { k++; } }
        if (k>m) { m=k; } }
    return(m); }
```

Question 12. Write a function *search* which computes for input y the largest integer x such that $x * x * x - 10000 * x \leq y$.

```
function search(y)
  { var x;
    x = 100;
    if (y>0) { x += Math.ceil(y); }
    while (x*x*x-10000*x > y) { x--; }
    return(x); }
```

Remark: No one got this right on all inputs. Therefore, all solutions were counted which are correct on those inputs y where y is a natural number.

Marksrange: 2–9 out of 0–12; most marks were 5,6,7.