Name	SOLUTION
Student ID #	
Instructor:	Sergey Kirshner

STAT 598G (Computational Statistics) Spring 2011
Midterm Exam
March 3, 2011

You are not allowed to use books or notes. Please read the directions carefully. There are 4 problems. The exam is graded out of **30**. If you complete all three parts of problem 2 (correctly), you can earn 15 points for that problem (3 points extra credit). Full credit is given for complete correct solutions. Partial credit is limited and awarded at the discretion of the instructor. You have **75** minutes to complete the exam. *Please show all your work.*

Problem #	Number of points
1	/6
2	/12
3	/6
4	/6
Total	/30

1. (6 points) Assume you are given an array $A[1,\ldots,n]$ of n integers. Provide an algorithm and its pseudocode for determining whether A contains two elements so that their difference is exactly 5. Analyze the algorithm and describe its computational complexity in the order of growth notation as a function of n.

   Solution: There are many possible solutions. Perhaps the simplest one is to consider each pair of elements and to check whether their difference is 5. Since there are $n(n-1)/2$ distinct pairs, the algorithm will have $O(n^2)$ running time complexity (e.g., Algorithm 1).

   There is a more elegant alternative. If an array is sorted in the increasing order, checking whether two elements are 5 apart can be done more efficiently by systematically checking the elements in increasing order. Sorting can be done in $O(n \log n)$ time, and checking can be done in $O(n)$ time, leading to the total $O(n \log n)$ running time algorithm (e.g., Algorithm 2).

Algorithm 1 Inefficient Solution

```
function SOLUTION1INEFFICIENT($A$)
    INPUTS: $n$-element array $A[1,\ldots,n]$
    OUTPUT: true if there exist $i$, $j$ so that $A[i] - A[j] = 5$; false otherwise.
    for $i = 1,\ldots,n-1$ do
        for $j = i+1,\ldots,n$ do
            if $|A[i] - A[j]| = 5$ then
                return true
            end if
        end for
    end for
    return false
end function
```

Algorithm 2 Efficient Solution

```
function SOLUTION1EFFICIENT($A$)
    INPUTS: $n$-element array $A[1,\ldots,n]$
    OUTPUT: true if there exist $i$, $j$ so that $A[i] - A[j] = 5$; false otherwise.
    $A =$MergeSort($A$)                    ▷ $O(n \log n)$ time
    $start = 1$                            ▷ index of the current element under consideration
    $end = 1$                              ▷ index for the furthermost element to consider
    while $end < n+1$ do                   ▷ no more elements to consider
        if $A[end] - A[start] = 5$ then
            return true                    ▷ done
        else if $A[end] - A[start] < 5$ then
            $end = end + 1$                ▷ elements 5 away from $A[start]$ have to be further in the array
        else                               ▷ $A[end] - A[start] > 5$
            $start = start + 1$            ▷ $A[start]$ cannot participate in such a pair
        end if
    end while
    return false
end function
```

2. **Generating random samples:** (12 points) For all of the questions below, assume you are given a random number generator `unif` () that draws samples from $\mathsf{Unif}\,(0,1)$, but none others. Assume you can perform standard mathematical operations (computing trigonometric functions, logs, powers, etc). Give a scheme for generating of random draws from the following random variables, and provide an explanation for why the proposed scheme is correct. Write down a pseudocode for drawing the samples. Be precise of all of the inputs, outputs, and constants used in the pseudocode.

   Each part is worth 6 points. You only need to solve two problems out of three for full credit. You can receive a maximum of 15 points for this problem.

   (a) **Log-normal:** X with a density $f_X(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{-\frac{(\ln x-\mu)^2}{2\sigma^2}}$, $x > 0$. Assume the parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$ are known. For partial credit, assume $\mu = 0$ and $\sigma^2 = 1$.

   Solution: If $Z \sim \mathcal{N}(0,1)$, then $X = e^{\sigma Z+\mu}$. To see this, notice that $Z = \frac{\ln X - \mu}{\sigma}$, and $\frac{\mathrm{d}Z}{\mathrm{d}X} = \frac{1}{\sigma X}$, so the density $f_X(x)$ would be

$$\begin{aligned} f_X(x) = f_Z(z)\left|\frac{\mathrm{d}z}{\mathrm{d}x}\right| &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\left(\frac{\ln x-\mu}{\sigma}\right)^2\right)\left|\frac{1}{\sigma x}\right| \\ &= \frac{1}{x\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2}\left(\frac{\ln x-\mu}{\sigma}\right)^2\right), \quad x > 0. \end{aligned}$$

   Z can be simulated using Box-Muller method, drawing $U_1, U_2 \overset{iid}{\sim} \mathsf{Unif}\,(0,1)$, and setting $Z = \sqrt{-2\ln U_1}\cos(2\pi U_2)$.

Algorithm 3 Sampling from log-normal

```
1: function SOLUTION2.A($\mu$,$\sigma$)
2:     INPUTS: parameters $\mu \in \mathbb{R}$, $\sigma > 0$
3:     OUTPUT: random draw $x \sim f_X$
4:     $u_1 = $ unif (), $u_2 = $ unif ()
5:     $z = \sqrt{-2\ln u_1}\cos(2\pi u_2)$
6:     $x = \exp(\sigma z + \mu)$
7:     return $x$
8: end function
```

(b) X with a pdf $f_X(x) \propto e^{-x^3},\ x > 0$.

Solution: We will have to use an Accept-Reject Algorithm. While there are many possible envelope functions, we will use $g_X(x) = e^{-x},\ x > 0$, primarily, because it is easy to draw $X \sim g_X$ ($X = -\ln(1-Y)$ for $Y \sim \mathsf{Unif}(0,1)$). Suppose $\tilde{f}_X(x) = e^{-x^3},\ x > 0$, and $C = \int_0^\infty e^{-x^3}\mathrm{d}x < \infty$, so that $f_X(x) = \frac{1}{C}e^{-x^3},\ x > 0$. We need to find MC so that $f_X(x) = \frac{1}{C}\tilde{f}_X(x) \le Mg_X(x),\ \forall x > 0$:

$$\operatorname*{argmax}_{x>0} \frac{\tilde{f}(x)}{g(x)} = \operatorname*{argmax}_{x>0} e^{-x^3+x} = \operatorname*{argmax}_{x>0} \left\{-x^3 + x\right\}.$$

By taking derivative of $-x^3 + x$, we find that the critical points are $x = 0$ and $x = \frac{1}{\sqrt{3}}$ with the latter serving as the argmax. So, by choosing $MC = \exp\left(-\left(1/\sqrt{3}\right)^3 + 1/\sqrt{3}\right)$, we guarantee that $f_X(x) \le Mg_X(x)\ \forall x > 0$. We can now set up an AR Algorithm by repeatedly drawing $x \sim g_X$ and $u \sim \mathsf{Unif}(0,1)$ until $u \le \frac{\tilde{f}_X(x)}{MCg_X(x)}$.

Algorithm 4 Sampling from e^{-x^3}

```
 1: function SOLUTION2.B
 2:     INPUTS:
 3:     OUTPUT: random draw $x \sim f_X(x) \propto e^{-x^3},\ x > 0$
 4:     $MC = \exp\left(-\left(\frac{1}{\sqrt{3}}\right)^3 + \frac{1}{\sqrt{3}}\right)$
 5:     repeat
 6:         $y = \texttt{unif}()$
 7:         $x = -\ln(1-u)$                          ▷ $x$ is a draw from $\mathsf{Exp}(1)$
 8:         $u = \texttt{unif}()$
 9:     until $u \le \exp\left(-x^3 + x\right)/MC$
10:     return $x$
11: end function
```

(c) (X,Y) with a joint pdf $f_{XY}(x,y) = 6x,\ 0 < x < y < 1$.

Solution: We will use a mixture approach, first generating from $y \sim f_Y$, and then $X \sim f_{X|Y}(\cdot|y)$. Both y and $x|y$ can be sampled using the inverse transform method.

$$\begin{aligned}
f_Y(y) &= \int_0^y 6x\,\mathrm{d}x = 3x^2\Big|_0^y = 3y^2,\ y \in (0,1),\\
F_Y(y) &= \int_0^y 3t^2\,\mathrm{d}t = y^3,\ y \in (0,1),\\
F_Y^{-1}(u) &= \sqrt[3]{u},\ u \in (0,1),\\
f_{X|Y}(x|y) &= \frac{f_{XY}(x,y)}{f_Y(y)} = \frac{6x}{3y^2} = \frac{2x}{y^2},\ x \in (0,y),\\
F_{X|Y}(x|y) &= \int_0^x \frac{2t}{y^2}\,\mathrm{d}t = \frac{x^2}{y^2},\ x \in (0,y),\\
F_{X|Y}^{-1}(u|y) &= y\sqrt{u},\ u \in (0,1).
\end{aligned}$$

So, one can first draw $y \sim f_Y$ using the inverse transform, and then sample $x|y \sim f_{X|Y}$ again using the inverse transform.

Algorithm 5 Sampling from $f_{XY}(x,y) = 6x,\ 0 < x < y < 1$

```
1: function SOLUTION2.C
2:     INPUTS:
3:     OUTPUT: random draw $(x,y) \sim f_{XY}(x,y) = 6x,\ 0 < x < y < 1$
4:     $u_1 = \texttt{unif}()$
5:     $y = \sqrt[3]{u_1}$
6:     $u_2 = \texttt{unif}()$
7:     $x = y\sqrt{u}$
8:     return $(x,y)$
9: end function
```

3. (6 points) The C program below is supposed to read in a 10×10 matrix and to output its trace. However, it contains several errors. List the errors and provide fixes to the problems. (Hint: there are four errors in total. Two of them will not let the program compile. One of them will cause a segmentation fault. One of them may lead to a wrong result.)

```c
#include <stdio.h>

float trace( float *A, int dim ) {
  // float tr;
  float tr=0.0;
  //
  int i;

  for( i=0; i<dim; i++ ) {
    // tr+=A[i][i];
    tr+=A[i*dim+i];
  }

  return( tr );
}

int main() {
  float A[10][10];
  int i,j;

  for( i=0; i<10; i++ ) {
    for( j=0; j<10; j++ ) {
      // scanf( "%f", A[i][j] );
      scanf( "%f", &A[i][j] );
    }
  }

  printf( "Trace of matrix A=%f\n", trace( (float*)&A[0][0], 10 ) );
  return( 0 );
}
```

Solution: The code above was modified. Incorrect lines were commented out, and the lines immediately below them indicate corrections. The code is available online.

4. (6 points) What is the output of the following program?

```
#include <stdio.h>
#include <stdlib.h>

void print1( int *A, int i ) {
  printf( "%d ", A[i] );
  i++;
  return;
}

void print2( int *A, int* i ) {
  printf( "%d ", A[*i] );
  (*i)++;
  return;
}

int main() {
  int i;
  int *A,*Ap;
  int B[10]={0,1,2,3,4,5,6,7,8,9};

  A=(int*)malloc(10*sizeof(int));
  Ap=A+10;
  for( i=0; i<10; i++ ) {
    Ap=Ap-1;
    *Ap=B[i];
  }

  for( i=0; i<10; i++ )
    print1( A, i );
  printf( "\n" );

  for( i=0; i<10; i++ )
    print2( A, &i );
  printf( "\n" );

  if( A<B )
    printf( "A is located before B in memory\n" );
  else
    printf( "A is located after B in memory\n" );

  free(A);
  return( 0 );
}
```

Solution: The program is available online – try it out! `A` is dynamically allocated, and is therefore in the heap segment, at the top part of the memory (earlier addresses); `B` is static, and is therefore resides on the stack, the bottom part of the memory (later addresses).

```
9 8 7 6 5 4 3 2 1 0
9 7 5 3 1
A is located before B in memory
```