C# Projekt 1 Name:

Statistische Untersuchungen zu endlichen Funktionsgraphen

Aufgabe: Basierend auf dem Abschnitt 2.1.6. „Random mappings“, Kap.2, S 54-55, in [1] sollen zunächst für eine beliebige Funktion $f:\{1,2,..,n\} \rightarrow \{1,2,...,n\}$ ($n \in \mathbf{N}$) C#-Routinen für die Berechnung der in Def. 2.35 eingeführten Kenngrößen des zugehörigen Funktionsgraphen geschrieben und damit die Fakten 2.37 und 2.38 für möglichst große Werte von n überprüft werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 2 Name:

Algorithmen zur Polynomarithmek über $\mathbf{Z}_p$

Aufgabe: Basierend auf den Abschnitten 2.6.2. und 2.6.3, Kap.2, S 81-85, in [1] sollen die dort angegebenen Algorithmen 2.221, 2.226 und 2.227 in C# implementiert und damit einige Beispiele gerechnet werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 3 Name:

Pollard's Rho und p – 1 Methode

Aufgabe: Basierend auf den Abschnitten 3.2.2. und 3.2.3, Kap.3, S 91-93, in [1] sollen für die dort angegebenen Faktorisierungsmethoden von Pollard entsprechende C#-Routinen geschrieben und ihre jeweiligen Vor- und Nachteile an gut ausgewählte Zahlenbeispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 4 Name:

Quadratisches Sieb

Aufgabe: Basierend auf dem Abschnitt 3.2.6., Kap. 3, S 95-97, in [1] soll das Quadratische Sieb (Algorithmus 3.21) zur Faktorisierung ganzer Zahlen in C# implementiert werden. Ferner soll auch anhand von Fallbeispielen untersucht werden, wie weit Bem. 3.24 für die gegebene Implementierung zutrifft, was die Größe der gewählten Faktorbasis angeht.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 5 Name:

Attacken auf DLP: Baby-step giant-step Algorithmus und Pollard's Rho Methode

Aufgabe: Basierend auf dem Abschnitt 3.6., Kap. 3, S 103-107, in [1] sollen die im Titel angesprochenen Algorithmen in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 6 Name:

Attacken auf DLP: Index-Calculus Algorithmus

Aufgabe: Basierend auf dem Abschnitt 3.6., Kap. 3, S 109-111, in [1] soll der Algorithmus 3.68 für den Spezialfall G= $\mathbf{Z}_p^*$ in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 7 Name:

Quadratfreie Faktorisierung von Polynomen über $\mathbf{Z}_p$

Aufgabe: Basierend auf dem Abschnitt 3.11., Kap. 3, S 122-123, in [1] soll der Algorithmus 3.110 in C# implementiert und an gut ausgewählten Beispielen demonstriert werden. (Dabei wird vereinfachend vorausgesetzt, dass der Koeffizientenring der Polynome der Restklassenring $\mathbf{Z}_p$ für eine Primzahl p ist.)

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 8 Name:

Probabilistische Primzahltests

Aufgabe: Basierend auf dem Abschnitt 4.2., Kap. 4, S 135-140, in [1] sollen die dort angeführten Algorithmen zu obigen Thema in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 9 Name:

Irreduzible Polynome über $\mathbf{Z}_p$

Aufgabe: Basierend auf dem Abschnitt 4.5., Kap. 4, S 154-160, in [1] sollen dort angegebenen Algorithmen zu obigem Thema in C# implementiert und an gut ausgewählten Beispielen (s. dazu auch die Tabellen auf S 158-159) demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 10 Name:

Das RSA-Verfahren und einige Attacken darauf

Aufgabe: Basierend auf dem Abschnitt 8.2, Kap. 8, S 285-291, in [1] soll das RSA-Verfahren in C# implementiert und einige mögliche Attacken darauf an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 11 Name:

Die Rabin-Variante von RSA

Aufgabe: Basierend auf dem Abschnitt 8.3, Kap. 8, S 292-294, in [1] soll die Rabin-Variante von RSA in C# implementiert und einige mögliche Attacken darauf an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 12 Name:

Das ElGamal Public-Key Kryptosystem

Aufgabe: Basierend auf dem Abschnitt 8.4, Kap. 8, S 294-296, in [1] soll das ElGamal Verschlüsselungsverfahren in seiner einfachsten Form, d.h. für die multiplikative Gruppe des Restklassenrings $\mathbf{Z}_p$, in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 13 Name:

Das Merkle-Hellman Knapsack-Verfahren

Aufgabe: Basierend auf dem Abschnitt 8.6, Kap. 8, S 300-302, in [1] soll das Merkle-Hellman Knapsack-Verfahren in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 14 Name:

Probabilistische Verfahren mit öffentlichem Schlüssel

Aufgabe: Basierend auf dem Abschnitt 8.7, Kap. 8, S 306-310, in [1] soll sollen die dort angegebenen Algorithmen zu obigen Thema in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 15 Name:

Algorithmen zur Berechnung des ggT(x,y)

Aufgabe: Basierend auf dem Abschnitt 14.4, Kap. 14, S 606-610, in [1] sollen die dort vorgestellten Algorithmen zu obigen Thema in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 16 Name:

Tests auf Zufälligkeit von Bitfolgen (1)

Aufgabe: Basierend auf dem Abschnitt 5.4.4, Kap. 5, S 181-183, in [1] soll eine Bitfolge einer vorgegebenen Länge n (z.B. n=512) mit Hilfe des C#-Implementierung eines Poker und Runs Tests auf Zufälligkeit getestet werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 17 Name:

Tests auf Zufälligkeit von Bitfolgen (2)

Aufgabe: Basierend auf dem Abschnitt 5.4.5, Kap. 5, S 183-184, in [1] soll eine Bitfolge einer vorgegebenen Länge n (z.B. n=512) mit Hilfe des Maurer Tests auf Zufälligkeit getestet werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 18 Name:

Tests auf Zufälligkeit von Bitfolgen (3)

Aufgabe: Basierend auf dem Abschnitt 5.5.1, Kap. 5, S 185-186, in [1] sollen mit Hilfe der dort angegebenen Algorithmen Bitfolgen erzeugt werden, die für kryptographiche Zwecke gut geeignet sind.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 19 Name:

Feige-Fiat-Shamir Signatur Schema

Aufgabe: Basierend auf dem Abschnitt 11.4.1, Kap. 11, S 447-449, in [1] soll das im Titel angesprochene Verfahren in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 20 Name:

Probabilistische Verschlüsselung von Blum-Goldwasser

Aufgabe: Basierend auf dem Abschnitt 8.7.2, Kap. 8, S 308-311, in [1] soll das im Titel angesprochene Verfahren in C# implementiert und an gut ausgewählten Beispielen demonstriert werden.

Zum Erstellen und Testen der Programme steht, soweit notwendig, der EDV-Raum 104 jeden Dienstag ganztägig zur Verfügung. Bei Fragen zu den Aufgaben wenden Sie sich bitte ausschließlich an den jeweiligen Übungsgruppenleiter in deren Sprechstunden.

[1] Menezes-Oorschot-Vanstone, Handbook of Applied Cryptography,
CRC Press,1996 (http://www.cacr.math.uwaterloo.ca/hac/)

C# Projekt 21 Name:

Algorithmen der Graphentheorie: Bestimmung von Minimalgerüsten

Aufgabe: Gegeben sei ein zusammenhängender ungerichteter Graph G=/(V,E) mit der Knottenmenge V und der Kantenmenge E zusammen mit einer Gewichtsfunktion w, welche jeder Kante e in E eine positive reelle Zahl w(e) zuordnet. Für verschiedene Anwendungen ist man speziell an sog. Minimalgerüsten für G interessiert, das sind spannende Bäume mit minimalem Gesamtwicht (=Summe der Gewichte aller darin vorkommenden Kanten). Die wichtigsten Algorithmen zur Bestimmung eines Minimalgerüsts sind nach Kruskal bzw. Prim benannt.

Man mache sich im Internet diesbezüglich kundig, z.B.

http://de.wikipedia.org/wiki/Algorithmus_von_Kruskal

http://de.wikipedia.org/wiki/Algorithmus_von_Prim

http://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Lehre/WS0910/Algo/folien3.pdf

und löse insbesondere dann die Aufgaben, welche verbunden sind mit

- der Eingabe des Graphen zusammen mit den Gewichten der Kanten
- seiner optischen Darstellung auf dem Bildschirm
- der Durchführung der Algorithmen zur Bestimmung eines Minimalgerüsts
- der optischen Kennzeichnung des Minimalgerüsts

Wenn möglich, sollte man dabei den Aufbau des Minimalgerüsts schrittweise mitverfolgen können.

C# Projekt 22 Name:

Algorithmen der Graphentheorie: Bestimmung von kürzesten Wegen

Aufgabe: Gegeben sei ein (gerichtetet oder ungerichteter) Graph G=(V,E) mit der Knotenmenge V und der Kantenmenge E zusammen mit einer Gewichtsfunktion w, welche jeder Kante e in E eine positive reelle Zahl w(e) zuordnet. Für verschiedene Anwendunghen (z.B. Routenplaner) sind die nun die (in Bezug auf die Gewichtsfunktion w) „kürzesten“ (und in gerichteten Graphen auch gerichteten) Wege von einem fest ausgewählten Startknoten zu jedem anderen Knoten interessant sind. Die wichtigsten Algorithmen dafür stammen von Dijkstra vzw. von Warshall-Floyd (letzterer vor allem dann, wenn die Aufgabe für jeden Startknoten durchgeführt werden soll).

Man mache sich im Internet diesbezüglich kundig, z.B.

http://de.wikipedia.org/wiki/Dijkstra-Algorithmus

http://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.190/Lehre/WS0910/Algo/folien3.pdf

und löse insbesondere dann die Aufgaben, welche verbunden sind mit

- der Eingabe des Graphen zusammen mit den Gewichten der Kanten
- seiner optischen Darstellung auf dem Bildschirm
- der Durchführung der Algorithmen zur Bestimmung der kürzesten Wege
- der optischen Kennzeichnung der kürzesten Wege zu einem bestimmten Endknoten
-

Wenn möglich, sollte man die Algorithmen schrittweise mitverfolgen können.

C# Projekt 23 Name:

Methode des kritischen Pfades (CPM)

Aufgabe: Für sog. Netzpläne, d.h., gerichtete, azyklische und schwach zusammenhängende Graphen mit einer Quelle und einer Senke und einer reellen und positiven Bewertungsfunktion w soll die CPM (=Critical Path Method) implementiert werden, wonach man dann für ein vorgegebenes Projekt, dessen Einzelstadien durch einen Netztplan repräsentiert werden, insbesondere für jedes Zwischenstadium der frühest- und der spätestmögliche Termin ersichtlich werden.

Man mache sich im Internet diesbezüglich kundig, z.B.

http://de.wikipedia.org/wiki/Methode_des_kritischen_Pfades

und löse insbesondere dann die Aufgaben, welche verbunden sind mit

- der Eingabe des Graphen zusammen mit den Bewertung der Kanten
- seiner optischen Darstellung auf dem Bildschirm
- der Durchführung der CPM
- der optischen Kennzeichnung von kritischen Pfaden

Wenn möglich, sollte man die Algorithmen schrittweise mitverfolgen können.

C# Projekt 24 Name:

Algorithmen der Graphentheorie: Euler- und Hamiltontouren

Aufgabe: Gegeben sei ein (gerichtetet oder ungerichteter) Graph G=(V,E) mit der Knotenmenge V und der Kantenmenge E. Eine Euler- bzw. Hamiltontour ist dann eine geschlossene Kantenfolge, welche jede Kante von E bzw. jeden Knoten von V genau einmal enthält. Speziell für Eulertouren gibt es sehr ute Möglichkeiten, schnell festzustellen, ob eine solche überhaupt exisitert und ggf. eine zu finden (Hierholzer-Algorithmus). Für Hamiltontouren sieht es diesbezüglich nicht so gut aus, für kleine Graphen sollte aber ein „intelligentes“ Backtracking-Verfahren noch zum Ziel führen, das von einem beliebigen Knoten startet und bei „Sackgassen“ zum letzten noch nicht voll abgearbeitet Knoten zurückkehrt.

Man mach sich im Internet kundig, z.B.

http://de.wikipedia.org/wiki/Algorithmus_von_Hierholzer

http://de.wikipedia.org/wiki/Eulerkreisproblem

http://de.wikipedia.org/wiki/Hamiltonkreisproblem

und löse insbesondere dann die Aufgaben, welche verbunden sind mit

- der Eingabe des Graphen
- seiner optischen Darstellung auf dem Bildschirm
- der Durchführung der Algorithmen
- der optischen Kennzeichnung von Euler- bzw. Hamiltontouren

Wenn möglich, sollte man die Algorithmen schrittweise mitverfolgen können.

C# Projekt 25 Name:

Traveling Salesman Problem (TSP)

Gegeben sei ein (gerichtetet oder ungerichteter) Graph G=(V,E) mit der Knotenmenge V und der Kantenmenge E zusammen mit einer Bewertungsfunktion w, welche jeder Kante e in E eine positive reelle Zahl w(e) zuordnet. Besitzt dann der Graph. Hamiltontouren, d.h., geschlossene Kantenfolgen, welche jeden Knoten von V genau einmal enthalten, so soll diejenige mit kleinster Gesamtbewertung gefunden werden.

Man mache sich im Internet diesbezüglich kundig, insbesondere was Branch- and Boundalgorithmen zur Lösung dieses Problems betrifft, z.B.

http://de.wikipedia.org/wiki/Problem_des_Handlungsreisenden

http://www.youtube.com/watch?v=-cLsEHP0qt0&feature=relmfu

und löse insbesondere dann die Aufgaben, welche verbunden sind mit

- der Eingabe des Graphen zusammen mit den Bewertung der Kanten
- seiner optischen Darstellung auf dem Bildschirm
- der Durchführung des Branch and Bound Algorithmus
- der optischen Kennzeichnung der optimalen Hamiltontour

Wenn möglich, sollte man den Algorithmus schrittweise mitverfolgen können.