

Protokoll zur Diplomprüfung in der theoretischen Informatik

Prüfer : J. Hromkovic
Prüfung: 30.09.2002
Dauer: 1h 15 min
Fächer: :
Effiziente Algorithmen (4 Std.)
Approximative und Randomisierte Algorithmen (4 Std.)
Compilerbau (4 Std.)
Note : 2.0

Effiziente Algorithmen

1. Dynamisches Programmieren
 - a. Erklären
 - b. Unterschied zu Teile und hersche
 - c. Matrixmultiplikation erklären und die Formel für die Rekursion erklären
2. NP Vollständigkeit:

Hromkovic: Sie können alle Reduktionen?

Ich: Ja

Hromkovic: Welche war schwer für Sie?

Ich: Alle sind einfach, wenn man die Idee kennt?

Hromkovic: Welche gefällt Ihnen?

Ich: Ich finde Färbbarkeit interessant.

Hromkovic: Dann erklären Sie...

K-COL \leq SAT – Konstruktion und Umgangssprachliche Begründung

3. Pseudopolynomiale Algorithmen
 - a. Die Idee erklären für Zahlenprobleme
 - b. Beispiel: Simple Knapsack
 - Definition, Idee
 - Algorithmus komplett beschreiben
 - Komplexität begründen
 - Wichtig ist die Begründung der Schätzung der oberen Schranke von $|TUPLE(i)|$

Approximative und Randomisierte Algorithmen

1. Was ist ein Approximationsschema?
2. Metrisches TSP und Christofides Algorithmus
 - a. Konstruktion und die Begründung einzelner Schritte
 - b. Beweis der Approximationsgüte
 - c. Wichtig: Alle schritte müssen BEGRÜNDET werden!!!
 - d. Wichtig: Dieses Frage hat großen Wert. Sie Überschattet die vorherige Leistung enorm. Daher sollte sie wirklich SITZEN!
3. Approximationsschema für Simple Knapsack
 - a. Modified Greedy SK Algorithm
 - b. Approximationsgüte beweisen
4. Primzahltest – SSA
 - a. Satz über die Anzahl von Nicht-Zeugen für ungerade Zahlen n mit ungeradem $(n-1)/2$
 - b. Beweis:
 - i. Die injektive Funktion
 - ii. Wahl von b für $n = p * q$ (Chinesischer Restsatz)

Compilerbau

1. CYK Algorithmus – was ist das, wie funktioniert
2. LL(k) Grammatik formal aufschreiben

Kommentar: Man sollte alle Beweise lernen. Alle Konstruktionen sollten begründet werden wenn nötig (z.B. Christofides Algorithmus). Bei den Beweisen war es folgendermaßen. Zuerst fragte prof. Hromkovic ob ich das oder dies beweisen kann. Ich habe dann zuerst die Idee von dem jeweiligen Beweis erklärt. Prof. Hromkovic wollte die wichtigen Schritte wissen. Beispielsweise die Definition der injektiven Abbildung bei dem Satz zum Primzahltest. Sind diese Punkte erklärt fragte er nach dem Rest des Beweises nicht.

Zu der Note: Ich habe es mir leider mit dem Christofides Algorithmus versaut. Dieser war ihm wichtig. Bei EffAlg kommen immer fragen zu Approximationsschemata. Metrisches TSP ist wohl beliebte Frage die man besser beantworten sollte.

Atmosphäre recht angenehm und ohne Stress. Wenn es hackt, versucht prof. Hromkovic mit fragen zu helfen. Ich habe mir oft Zeit genommen um kurz nachzudenken. Top Tipp: Zuerst sollte man das prüfen lassen, wo man gut Punkten kann. Denn ein guter erster Eindruck ist am wichtigsten.

That's all folks.

Protokoll zur Diplomprüfung in theoretischer Informatik

Prüfer : J. Hromkovic

Prüfung : 16.05.2002

Dauer : 35 min

Fächer :

Alg. Kryptographie (4 Std.)

Effiziente Algorithmen (4 Std.)

Compilerbau (4 Std.)

Note : 1.0

Kryptographie

1. RSA
 - Aufbau
 - Was ist Φ von n ?
 - Beweis: Eindeutigkeit alle drei Fälle
2. Alle Protokolle nennen
 - Einfache digitale Unterschrift
 - Digitale Unterschrift : El Gamal
3. Zero Knowledge Proof
 - Definition
 - 3-Färbbarkeit

Effiziente Algorithmen

1. Par-Parametrisierung
 - Definition
 - Beispiel :
 - Vertex Cover (Greedy Methode) & Komplexität
2. NP-Vollständigkeit
 - SAT auf CLIQUE
3. Approximations-Algorithmus PTAS
 - Definition
 - Beispiel :
 - Einfache Rucksackproblem , Komplexität & Beweis von Güte
3. Metrische TSP
 - Beweis & Komplexität

Compilerbau

1. Was ist ein Compiler?
2. Phasen
3. Methoden zur Syntaktische Analyse
 - Grobe Beschreibung von :
 - Top-Down-Methoden
 - Bottom-Up-Methoden
 - Unvisersaler- Parser-Methoden
4. LL(k)
Defintion und Formale Schreibweise.

That's all