Institut für Computergraphik und Algorithmen

Abteilung für Algorithmen und Datenstrukturen

TECHNISCHE UNIVERSITÄT WIEN

Prof. Petra Mutzel
Günther Raidl
Gunnar Klau
Gabriele Kodydek
René Weiskircher

Wintersemester 2001/2002

Prüfung zur Vorlesung Algorithmen und Datenstrukturen 2 März 2002

a) Machen Sie bitte die folgenden Angaben in deutlicher Blockschrift:

Vorname: _____ Nachname: _____

Matrikelnummer: _____ Studienkennzahl: _____

b) Legen Sie während des Tests Ihren Studentenausweis vor sich auf das Pult.

c) Schreiben Sie die Lösungen direkt auf das jeweilige Aufgabenblatt. Wenn Ihnen das Papier ausgeht, bitten Sie die Aufsicht um Nachschub. Es ist nicht erlaubt, eigenes Papier zu verwenden!

d) Denken Sie daran, dass keinerlei Hilfsmittel erlaubt sind – weder Taschenrechner, irgendwelche Unterlagen, Mobiltelefone etc.

VOR DER ABGABE AUSZUFÜLLEN:

Geben Sie bitte die Anzahl der zusätzlich abgegebenen Blätter an: _____

Resultat:

Aufgabe	A 1	A 2	A 3	A 4	A5	
maximale Punktzahl	10	10	10	10	10	50
erreicht						

Viel Erfolg!

Aufgabe 1.A: Graphenalgorithmen **(10 Punkte)**

Beschreiben Sie ein Verfahren, mit dem man feststellen kann, ob ein gegebener Graph $G = (V, E)$ ein Baum ist. Ihr Algorithmus soll in Zeit $O(|V|)$ laufen.

a) (3 Punkte)
Beschreiben Sie die Idee Ihres Algorithmus.

b) (3 Punkte)
Geben Sie kommentierten Pseudocode an.

c) (2 Punkte)
Analysieren Sie die Laufzeit Ihres Verfahrens.

d) (2 Punkte)
Beweisen/argumentieren Sie, warum Ihr Algorithmus korrekt ist.

Aufgabe 2.A: Evolutionäre Algorithmen – MCPP **(10 Punkte)**

Das Multiple Container Packing Problem (MCPP) ist ein kombinatorisches Optimierungsproblem, das folgendermaßen definiert ist: Gegeben seien n Gegenstände mit gegebenen Gewichten w_j und Werten v_j für $j = 1, \ldots, n$ sowie C Behälter, deren Kapazität jeweils durch das zulässige Gesamtgewicht $W_{\max}$ beschränkt ist. Gesucht wird eine Zuordnung der Gegenstände zu den Containern, die einen maximalen Gesamtwert der zugeordneten Gegenstände ergibt, unter der Bedingung, dass für keinen Behälter das zulässige Gesamtgewicht überschritten wird. Gegenstände, die aus Platzmangel keinem Behälter zugeordnet werden können, tragen also auch zum Gesamtwert nichts bei.
Das Problem soll durch einen evolutionären Algorithmus gelöst werden. Eine Lösung sei indirekt durch eine Permutation aller Gegenstände $j = 1, \ldots, n$ repräsentiert, die in einem Feld B der Länge n gespeichert sind (Order Based Encoding). Die Zuordnungen der Gegenstände zu den Behältern werden daraus mittels einer First-Fit-Heuristik berechnet: alle Behälter i ($i = 1, \ldots, C$) werden in der Reihenfolge mit Gegenständen gefüllt, wie sie durch die Permutation gegeben sind. Wenn Gegenstand $B[t]$ nicht mehr in den aktuellen Behälter i passt, wird er in den nächsten Container $i + 1$ gepackt und der Algorithmus fährt fort, diesen Container zu befüllen. Wenn alle Behälter auf diese Art gefüllt sind, werden die übrigen Gegenstände als nicht zugewiesen behandelt.

a) (3 Punkte)
Erzeugt diese First-Fit-Heuristik auf der vorgestellten Kodierung nur zulässige oder auch unzulässige Lösungen? Begründen Sie Ihre Antwort mit wenigen Worten.

b) (3 Punkte)
Erklären Sie die Aufgabe der Rekombination in einem evolutionären Algorithmus. Beschreiben Sie Uniform Crossover und erläutern Sie, ob diese Methode für das Multiple Container Packing Problem mit Order Based Encoding geeignet ist.

c) (4 Punkte)
Beschreiben Sie mit wenigen Worten einen Mutationsoperator, der für das Multiple Container Packing Problem mit Order Based Encoding geeignet ist.

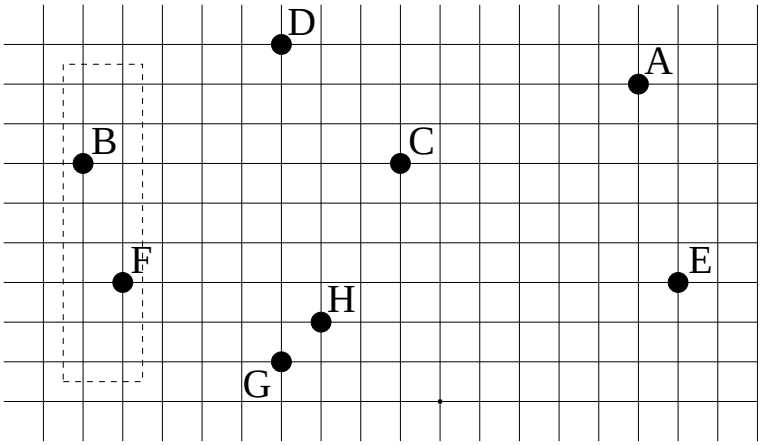
Geben Sie dann einen Pseudocode für diesen Mutationsoperator an, der aus einer Lösung B eine mutierte Lösung M erzeugt. Verwenden Sie dazu die Funktion `Random()`, welche als Ergebnis eine gleichverteilte Zufallszahl aus dem Intervall $[0, 1)$ zurückliefert.

Aufgabe 3.A: Geometrische Algorithmen **(10 Punkte)**

Gegeben ist die unten abgebildete Konfiguration von Punkten.

a) (5 Punkte)
Zeichnen Sie einen balancierten zweidimensionalen Suchbaum für diese Konfiguration. Schreiben Sie dabei zu jeder Ebene des Baums dazu, ob die Söhne der Knoten dieser Ebene durch ihre x– oder y–Koordinate bestimmt sind.

b) (5 Punkte)
Welche Knoten des Baums müssen Sie bei der Bereichsabfrage nach dem mit gestrichelten Seiten gezeichneten Rechteck betrachten? Zählen Sie alle Knoten auf, die der Bereichssuche–Algorithmus betrachtet.



Aufgabe 4.A: Dynamische Programmierung **(10 Punkte)**

Der Binomialkoeffizient

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

gibt die Anzahl der k–elementigen Teilmengen einer n–elementigen Menge an. Um einen Algorithmus zur Berechnung von Binomialkoeffizienten anzugeben, kann man das *Pascal–Dreieck* benutzen, das auf folgendem Satz beruht.
Für alle $n, k \in \mathbb{N}$ mit $n > k$ gilt:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k} \ .$$

Diese Formel legt einen einfachen rekursiven Algorithmus nahe, der allerdings exponentielle Laufzeit hat.

a) (6 Punkte)
Geben Sie einen Algorithmus `binomial`(n, k) an, der den Binomialkoeffizienten $\binom{n}{k}$ berechnet und der auf dem Prinzip der dynamischen Programmierung beruht. Hinweis: Es bietet sich an, ein zweidimensionales Feld b zu verwenden mit $b[i, j] = \binom{i}{j}$ und dieses für $j = 0$ mit dem Wert 1 zu initialisieren.

b) (2 Punkte)
Analysieren Sie die Laufzeit Ihres Verfahrens.

c) (2 Punkte)
Worin besteht der Vorteil gegenüber der rekursiven Berechnung?

Aufgabe 5.A: Suchen in Texten **(10 Punkte)**

Wie aus der Vorlesung bekannt, geht es beim *Pattern Matching Problem* darum, Vorkommen eines Musters P der Länge M in einem Text T der Länge N zu finden.

a) (2 Punkte)
Beschreiben Sie die Idee des naiven Verfahrens aus der Vorlesung und geben Sie dessen Laufzeit an.

b) (6 Punkte)
Nehmen Sie nun an, dass alle Zeichen in P voneinander verschieden sind. Beschreiben Sie unter Zuhilfenahme von kommentiertem Pseudocode, wie Sie den naiven Algorithmus ändern müssen, damit er in Zeit $O(N)$ läuft.

c) (2 Punkte)
Visualisieren Sie einen Aufruf Ihres Verfahrens am Beispiel

$$T = \text{I M M A E R Z E N D E R B A U E R}$$
$$P = \text{D E R B}$$