



19. Bundeswettbewerb Informatik 2000/2001

Die Aufgaben der 2. Runde

Allgemeine Hinweise

An dieser Runde können nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben mindestens 12 Punkte erreicht haben oder einer Gruppe angehört haben, der dieses gelungen ist. Gruppenarbeit ist in der zweiten Runde nicht mehr zulässig. Einsendeschluss ist der **23. April 2001**, wobei das Datum des Poststempels gilt. Bitte adressieren Sie Ihre Einsendungen wieder an den

Bundeswettbewerb Informatik, Ahrstraße 45, 53175 Bonn

Es gibt drei Aufgaben; alle müssen bearbeitet werden. Wir erwarten zunächst eine einfache, nachvollziehbare und vollständige Lösung aller Teilaufgaben inklusive der geforderten Beispiele (teilweise in der Aufgabenstellung vorgegeben). Eine höhere Bewertung auf der nach oben offenen Bewertungsskala können Sie erreichen, wenn Sie die Aufgaben an Stellen, an denen dies möglich und sinnvoll ist, eigenständig weiterentwickeln. Dabei bevorzugen wir inhaltliche Erweiterungen und Verbesserungen, etwa von Datenstrukturen und Algorithmen, und legen keinen Wert auf aufwendige Tricks, z.B. zur reinen Verschönerung der Mensch-Computer-Interaktion oder der grafischen Ausgabe. Begründen Sie für jede Erweiterung, warum sie sinnvoll ist.

Denken Sie bitte daran, dass zur Bewertung möglicherweise nur Ihre Papier-Unterlagen herangezogen werden können. Diese sollten also u.a. ein lückenloser und nachvollziehbarer Nachweis des Leistungsumfangs und der Funktionstüchtigkeit Ihrer Programme sein. Senden Sie uns bitte Ergebnisse von Programmläufen mit unterschiedlichen Daten. Dem Umfang Ihrer Einsendung setzen Sie bitte Grenzen, denn eine gute Dokumentation vermittelt kurz aber präzise alles Notwendige, insbesondere Ihre wesentlichen Lösungsideen. Überlange Dokumentationen sind schon deshalb zu vermeiden, weil bei der Korrektur im ungünstigsten Fall positive Punkte darin übersehen werden können. Die Betonung liegt aber auf "präzise", denn eine kurze Dokumentation, die nicht die nötigen Informationen liefert, ist natürlich nutzlos. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die nur die Aufgabenstellung kennen, wissen müssen, um Ihre Lösungsidee zu verstehen und Ihre Realisierung dieser Idee nachzuvollziehen. Generell sind gute und originelle Ideen zwar entscheidend, aber die Dokumentation hat schon oft den Ausschlag für oder gegen das Weiterkommen gegeben.

Grundsätzlich gelten die Gliederungs- und Dokumentationsrichtlinien der 1. Runde weiter. Zu jeder Teilaufgabe gehört also die Lösungsidee und die Dokumentation der Lösung (eine Beschreibung, wie die Idee z.B. in eine Datenstruktur umgesetzt oder im Programm realisiert wurde, Hinweise auf Nutzungsgrenzen, Besonderheiten usw.). Bei Teilaufgaben ohne Programmierung verwenden Sie (halb-)formale Notationen, z.B. zur Beschreibung von Datenstrukturen oder Funktionen, geben Sie aber (wenn nicht ohnehin in einer anderen Teilaufgabe gefordert) auch eine Umsetzung in einer Programmiersprache an. Für die geforderten Programme erwarten wir zusätzlich ein Programmablaufprotokoll (kommentierte Probeläufe des Programms) und den Programmtext.

Schicken Sie uns alles in lesbarer Form auf Papier. Außerdem senden Sie uns bitte die Programmtexte und lauffähigen Programme auf einer **3,5"-Diskette oder CD** (lesbar unter DOS/Windows95). Falls für grafische Ausgaben keine Druckfunktion vorhanden ist, tut es ein Foto des Bildschirminhaltes auch. Bitte schicken Sie uns Ihre Unterlagen auf einzelnen, **gelochten** Blättern im Format DIN A 4 (**ohne Heftstreifen, Klarsichthüllen, Mappen o.ä.!**)



und geben Sie auf jedem Blatt Verwaltungsnummer, Vorname, Name und Seitennummer an. Sie finden die Verwaltungsnummer auf Ihrer Teilnahmebescheinigung der ersten Runde. Bitte gliedern Sie Ihre vollständige Einsendung in (a) Allgemeines, (b) Unterlagen zur Aufgabe 1, (c) Unterlagen zur Aufgabe 2 und (d) Unterlagen zur Aufgabe 3.

Allen Teilnehmern der zweiten Runde wird Anfang Juni die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die vom 2. bis 5. Oktober 2001 in Frankfurt stattfinden wird. Dort werden die Bundessieger und Preisträger ermittelt und am letzten Tag bei einer Siegerehrung ausgezeichnet.

Der Rechtsweg ist wie üblich ausgeschlossen.

Für Fragen zu den Aufgaben gibt es wieder eine Hotline, und zwar unter 0228 - 37 86 46 jeweils von 17-19h am 8.1., 14.2., 19.3. und 19.4. 2001 oder unter bwinf@bwinf.de. Bitte rechnen Sie bei E-mail-Anfragen mit leicht verzögerten Antworten. Informationen zur 2. Runde finden sich wieder auf unseren Webseiten unter www.bwinf.de. In der newsgroup fido.ger.bwinf wird sicher wieder über die Aufgaben diskutiert werden – aber Lösungs-ideen dürfen nicht ausgetauscht werden.

Auf einige Dinge möchten wir Sie noch hinweisen, die Sie vielleicht zusätzlich motivieren können, sich an der zweiten Runde aktiv zu beteiligen. Das Land Nordrhein-Westfalen wird die besten Zweitrundenteilnehmer aus NRW mit einem Buchpreis belohnen. Auch in anderen Bundesländern gibt es Anerkennungen für gute Teilnehmer an der zweiten BWINF-Runde. In allen Ländern ist es außerdem prinzipiell möglich, eine Einreichung zur zweiten Runde des Bundeswettbewerbs Informatik als so genannte besondere Lernleistung in die Abiturwertung einfließen zu lassen. Genauer erfahren Sie an Ihrer Schule.

Viel Spaß und viel Erfolg!

Aufgabe 1 DNA-Sequenzierung nach der "shotgun"-Methode

Die Desoxyribonukleinsäure (abgekürzt DNA) kommt in jeder lebenden Zelle als Träger der Erbinformation vor. Die DNA ist eine sehr lange Folge von vier Grundbestandteilen, den Nukleotiden Adenin, Cytosin, Guanin und Thymin (abgekürzt A, C, G und T).

Das Human-Genom-Projekt hat sich zur Aufgabe gestellt, die menschliche DNA zu sequenzieren, d.h. die gesamte Folge der Nukleotide zu bestimmen. In chemischen Experimenten kann man kürzere Folgen von Nukleotiden lesen.

Ein DNA-Molekül kann durch einen String, zusammengesetzt aus den Buchstaben A, C, G und T, beschrieben werden. Ein Beispiel:

AACCTGATGGGTACTAAGTCGTAACCGTTGCATGTACGTTA... (1)

DNA-Moleküle können durch Enzyme aufgebrochen werden. Ein Enzym-Molekül kann immer nur an einer für das Enzym charakteristischen Nukleotidfolge "andocken" (beispielsweise ATG). Die DNA wird dann an einer bestimmten Stelle innerhalb dieser Folge getrennt (etwa zwischen A und T). Dieses Beispiel-Enzym könnte durch A | TG beschrieben werden. Die Zerlegung des DNA-Strings (1) durch dieses Enzym liefert folgende Teilstrings:

AACCTGA , TGGGTACTAAGTCGTAACCGTTGCA und TGTACGTTA... (2)

Ein anderes Enzym könnte etwa durch AAC | C beschrieben werden. Dieses zerlegt den obigen DNA-String in die Teilstrings

AAC , CTGATGGGTACTAAGTCGTAAC und CGTTGCATGTACGTTA... (3)



Bei einer Methode zur Sequenzierung von DNA, der „shotgun“-Methode, verfährt man wie folgt:

Zunächst werden durch Replikation sehr viele identische DNA-Moleküle bereitgestellt, die nachfolgend für verschiedene Experimente verwendet werden. Im ersten Experiment wird DNA mittels des ersten Enzyms zerschnitten. Da die entstehenden Bruchstücke nun überschaubare Länge haben, ist es möglich, deren Nukleotidfolge zu analysieren. In einem weiteren Experiment wird eine weitere Probe der ursprünglichen DNA mit einem anderen Enzym aufgetrennt und wieder die Nukleotidfolge der entstehenden Teile analysiert. Das Sequenzierungsproblem besteht nun darin, aus diesen Mengen von Teilstücken (wie etwa in (1) und (2) oben dargestellt) auf die Nukleotidfolge der gesamten DNA zu schliessen.

Die **Aufgabe** besteht darin, diesen Vorgang im Computer nachzubilden.

1. Erzeuge per Zufall einen DNA-String der Länge 10000.
2. Erzeuge per Zufall zwei verschiedene Enzymbeschreibungen der Länge 3 oder 4, wobei die Trennstelle (|) ebenfalls zufällig festgelegt wird. Dabei soll die Enzymbeschreibung der Einfachheit halber aus mindestens drei verschiedenen Buchstaben bestehen.
3. Zerlege in diesem Modell die DNA, einmal mittels des ersten, einmal mittels des zweiten Enzyms.
4. Entwickle einen Algorithmus, der anhand der Teilstrings aus beiden Experimenten den gesamten DNA-String rekonstruiert.
5. Vergleiche den gefundenen String aus 4 mit dem ursprünglichen aus 1.

Aufgabe 2 Screen.Dance

In der ersten Runde dieses Wettbewerbs gab es eine Aufgabe "Night.Dance" (Nummer 5). Dabei ging es darum, unter Berücksichtigung von sechs Regeln taktweise eine Choreografie zu erstellen.

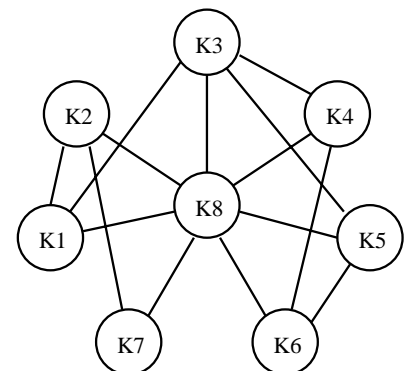
In der zweiten Runde soll es nun um Choreografie-Strukturen gehen und um die Visualisierung von Choreografie-Abläufen und Choreografie-Strukturen.

Damit sich auch interessante Strukturierungsmöglichkeiten öffnen, verzichten wir im Weiteren auf die letzten beiden der sechs Regeln:

- Jeder Tänzer hat nach seinem Sprung mindestens einen Takt lang eine Pause.
- In jedem Takt sollen sich möglichst viele Tänzer bewegen.

Es bleiben also die folgenden Regeln gültig:

- In einigen, aber nicht in allen Kreisen steht ein Tänzer.
- Tänzer springen immer zu Beginn eines Taktes.
- Ein Tänzer kann sich in einem gigantischen Sprung von seinem Kreis entlang einer Verbindungslinie in einen anderen Kreis bewegen, der schon zu Beginn des Taktes frei sein muss.
- Die während eines Taktes benutzten Verbindungslinien dürfen sich nicht kreuzen, um blutige Zusammenstöße in der Luft zu vermeiden.



Die Bühnenmarkierung mit ihren Kreisen und Verbindungslinien bleibt unverändert (s. Bild).



Aufgabenteil 1

Schreiben Sie ein Programm, das nach den vier noch gültigen Regeln eine Choreografie erstellt, wobei deren Länge sowie Anzahl und Position der Tänzer frei wählbar sind.

Erweitern Sie das Programm so, dass es den Ablauf der Choreografie direkt auf dem Computerbildschirm visualisiert. Die Qualität dieser Visualisierung hängt davon ab, ob man beim Zuschauen erkennen kann, wie der Algorithmus zur Erstellung der Choreografie seine Entscheidungen trifft.

Bei der Entwicklung der Visualisierung treffen Sie vermutlich eine ganze Reihe von Design-Entscheidungen. Listen Sie uns möglichst viele dieser Entscheidungen auf, jeweils mit einer sehr kurzen aber schlüssigen Begründung für diese Entscheidung.

Aufgabenteil 2

Das Programm aus dem Aufgabenteil 1 soll so erweitert werden, dass es Choreografien erstellt, denen Strukturen zugrundeliegen. Damit meinen wir, ähnlich zu Strukturen in Musikstücken, z.B. die Wiederholung von Taktsequenzen und das Vorkommen von Symmetrien.

Überlegen Sie sich mögliche Strukturelemente und entwickeln Sie ein eingebaues und druckbares symbolisches Format zur Beschreibung Ihrer Strukturelemente und Choreografie-Strukturen.

Erweitern Sie das Programm aus Aufgabenteil 1 so, dass es anhand einer vorgegebenen Strukturbeschreibung (in dem von Ihnen entwickelten Format) eine dazu passende, nach den vier Regeln gültige Choreografie erstellt.

Erweitern Sie die Visualisierung aus Aufgabenteil 1 so, dass die Choreografiestruktur beim Ablauf mit sichtbar wird. Die Qualität dieser Visualisierung hängt davon ab, wie gut man die Strukturelemente und ihre Abfolge erkennen kann.

Bei dieser Erweiterung der Visualisierung um Strukturinformation treffen Sie vermutlich eine ganze Reihe weiterer Design-Entscheidungen. Listen Sie uns wiederum möglichst viele dieser Entscheidungen auf, jeweils mit einer sehr kurzen aber schlüssigen Begründung für diese Entscheidung.

Aufgabenteil 3

Schicken Sie uns drei Beispiele mit jeweils Strukturbeschreibung, Anzahl und Anfangspositionen der Tänzer, zu denen Ihr Programm jeweils eine Choreografie von maximal 100 Takten Länge erstellen und visualisieren kann.

Damit wir alle Ihre Visualisierungen problemlos auf unseren Computern anschauen und beurteilen können, beachten Sie bitte:

- Ihr Programm muss auf einer der bei uns installierten Betriebssystemversionen lauffähig sein, und zwar sind dies Standardinstallationen von Windows95 und Windows98, der Linux-Distribution SuSE 6.3 und von MacOS 9.0.4.
- Ihr Programm muss ausführbar sein, ohne dass vorher irgendwelche Installationen nötig sind.

Beachten Sie auch: Die Bewertung der Visualisierungen richtet sich nicht nach dem Aufwand für die grafische Gestaltung, sondern nach den in den Aufgabenteilen 1 und 2 genannten Kriterien.



Aufgabe 3 Wortstämme

Suchmaschinen suchen meistens nicht wirklich in den Dokumenten, in denen sie Stichwörter finden sollen. Vielmehr wird von Zeit zu Zeit eine Indexliste gebildet, in der mit geringerem Aufwand gesucht werden kann. Solche Indexlisten müssen möglichst klein sein und dennoch alles Wesentliche enthalten. Man kann den Beitrag eines Dokuments zu einer Indexliste dadurch erhalten, dass man alle in dem Dokument vorkommenden Wörter aufnimmt, dann aber alle unwesentlichen aussondert. Artikel und Konjunktionen beispielsweise sind sicher nicht von Bedeutung. Solche unbedeutenden Wörter werden üblicherweise in einer Stoppwortliste für eine Sprache erfasst.

Noch kürzere Listen entstehen, wenn man nur Wortstämme zulässt. Im Englischen kann man mit wenigen Regeln die meisten Wörter korrekt auf ihre Stämme zurückführen. Zwischen Groß- und Kleinschreibung soll nicht unterschieden werden. Das ' ist das englische Hochkomma. Alle anderen Nichtbuchstaben werden wie Leerzeichen (blank) behandelt.

Ersetzungsregeln für Wortendungen in englischen Texten zur Bildung von Wortstämmen (die Reihenfolge ist wichtig):

% bedeutet: a, e, i, o, u oder y

* bedeutet: ein beliebiger Konsonant.

Nr.	Wortendung	wird ersetzt durch bzw. weggelassen	Voraussetzung
1	ies'	y	
2	es'		
3	s'		
4	's		
5	'		
6	ies	y	
7	es		*o, ch, sh, ss, zz oder x gehen voraus
8	s		*, e, %y, %o, oa oder ea gehen voraus
9	ing		** , % oder x gehen voraus
10	ing	e	%* gehen voraus
11	ied	y	
12	ed		** , % oder x gehen voraus
13	ed	e	%* gehen voraus

Aufgabe:

- Schreiben Sie ein Programm, welches einen englischen Text einliest und eine möglichst kleine Indexliste dazu erstellt.
- Liefern Sie je eine Indexliste zu den drei Texten unter
 - <http://www.nara.gov/exhall/charters/declaration/declaration.html> (ohne die 56 signatures),
 - <http://www.shef.ac.uk/~gaipp/control1.html> (Abschnitte 1 bis 6) und
 - <http://alt.venus.co.uk/weed/writings/poems/wvtd.htm>.

Eine dazu passende Stoppwortliste wie auch noch einmal alle Texte finden Sie über <http://www.bwinf.de/aufgaben/runde2/runde2.php3>.

- Eignen sich Indexlisten obiger Art zur Suche in Dokumenten allgemein? Wie kann man die Qualität von Suchergebnissen verbessern?