

29. Bundeswettbewerb Informatik 2010/2011

Die Aufgaben der zweiten Runde

Allgemeine Hinweise

Herzlichen Glückwunsch zum Erreichen der zweiten Runde! Hier sind die Aufgaben. Sie sind anspruchsvoll, und ihre Bearbeitung ist aufwändig. Aber die Mühe lohnt sich, denn durch Teilnahme an der zweiten Runde

- wirst du sicher sehr viel lernen;
- kannst du dich für die Endrunde qualifizieren;
- wirst du als Endrundenteilnehmer vielleicht zu weiteren Veranstaltungen eingeladen;
- kannst du mit einer guten Leistung einen Buchpreis des Verlags O'Reilly gewinnen;
- hast du am Ende eine Arbeit fertig gestellt, die du als so genannte „Besondere Lernleistung“ in die Abiturwertung einbringen kannst;
- kannst du dich (als jüngerer Teilnehmer) um die Teilnahme an einer Deutschen Schülerakademie bewerben;
- hast du die Chance auf die Teilnahme an einem Seminar beim Max-Planck-Institut für Informatik in Saarbrücken.

Wir wünschen also viel Spaß und viel Erfolg bei der Bearbeitung!

Wichtig: An dieser Runde dürfen nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben mindestens 12 Punkte erreicht oder einer Gruppe angehört haben, der dieses gelungen ist. Gruppenarbeit ist in der zweiten Runde nicht zulässig.

Einsendeschluss ist der 26. April 2011; es gilt das Datum des Poststempels. Bitte sende deine Lösungen wieder an den **Bundeswettbewerb Informatik, Wachsbleiche 7, 53111 Bonn.**

Es gibt drei Aufgaben. **Wichtig:** Eine Einsendung darf nur Bearbeitungen zu höchstens zwei Aufgaben enthalten, deren Bewertung dann das Gesamtergebnis ausmacht. Sollte eine Einsendung Bearbeitungen zu allen drei Aufgaben enthalten, werden wir zwei davon zufällig auswählen und nur diese bewerten.

Die Bearbeitung einer Aufgabe sollte zunächst eine einfache, nachvollziehbare und vollständige Lösung aller Teilaufgaben enthalten. **Pluspunkte** für eine höhere Bewertung kannst du erreichen, wenn du die Aufgabe dort, wo es möglich und sinnvoll ist, eigenständig weiterentwickelst. Sinnvoll sind inhaltliche Erweiterungen und Verbesserungen, etwa von Datenstrukturen und Algorithmen; uninteressant sind aufwändige Tricks, z. B. zur reinen Verschönerung der Benutzungsoberfläche. Begründe für jede Erweiterung, weshalb sie sinnvoll ist und ihre Realisierung eine eigene Schwierigkeit darstellt.

Denke bitte daran, dass zur Bewertung möglicherweise nur die Papierunterlagen herangezogen werden können. Diese sollten also einen lückenlosen und nachvollziehbaren Nachweis des Leistungsumfangs und der Funktionstüchtigkeit der Programme geben. Der Umfang der Einsendung soll sich in Grenzen halten; eine gute Dokumentation vermittelt kurz und präzise alles Nötige, insbesondere die wesentlichen Lösungsideen. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die die Aufgabenstellung kennen, wissen müssen, um die Lösungsidee zu verstehen und die Realisierung dieser Idee nachzuvollziehen. Generell sind zwar gute und originelle Ideen 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 sowie des dazu gehörigen Programms (eine Beschreibung, wie die Idee in konkrete Programmelemente umgesetzt wurde, Hinweise auf Nutzungsgrenzen, Besonderheiten usw.). Dabei sind (halb-)formale Notationen besser als Programmausschnitte. Für die geforderten Programme erwarten wir Programmablaufprotokolle, also kommentierte Probeläufe des Programms, aus denen ersichtlich wird, wie das Programm sich in unterschiedlichen Situationen verhält. Sende uns außerdem bitte (abgedruckt!) aussagekräftige Ergebnisse von Programmläufen mit unterschiedlichen Daten, auch wenn Beispiele nicht explizit gefordert sind. Komplettiert wird das Papiermaterial durch den Programmtext, wobei unwichtige und automatisch generierte Teile nicht ausgedruckt werden sollen.

Schicke uns alles in lesbarer Form auf Papier, Schriftgröße mindestens 10 Punkt, bei Quelltext mindestens 8 Punkt. Verwende bitte lose, gelochte Blätter im Format DIN A4 (Hüllen mit Lochrand nur bei ausreichender Stabilität verwenden; keine Heftstreifen oder Mappen) und gib auf jedem Blatt Verwaltungsnummer, Vorname, Name und Seitennummer an. Die Verwaltungsnummer steht auf der Teilnahmebescheinigung der ersten Runde (dort versehentlich als Einsendungsnummer bezeichnet). Bitte gliedere deine Einsendung in (a) Allgemeines (z. B. Angaben zu technischen Voraussetzungen, persönliche Anmerkungen etc.), (b) Unterlagen zur ersten bearbeiteten Aufgabe und (c) Unterlagen zur zweiten bearbeiteten Aufgabe.

Außerdem sende uns bitte die Programmtexte und lauffähigen Programme auf einer CD oder DVD. Bei der Bewertung können Programme unter Windows (XP / Vista / 7), Linux und Mac OS X (10.6) ausgeführt werden.

Fragen zu den Aufgaben können per E-Mail an bwinf@bwinf.de oder telefonisch unter 0228-378646 (zu üblichen Arbeitszeiten) gestellt werden. Die Antwort auf E-Mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf unseren Webseiten (www.bundeswettbewerb-informatik.de). Auf einstieg-informatik.de werden in der Community sicher wieder viele Teilnehmer über die Aufgaben diskutieren – ohne Lösungsideen auszutauschen.

Allen Teilnehmern der zweiten Runde wird Anfang Juni die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die vom 20. bis zum 23. September 2011 vom Fachbereich Informatik der TU Braunschweig ausgerichtet werden wird. Dort werden die Bundessieger und Preisträger ermittelt und ausgezeichnet. Bundessieger werden in der Regel ohne weiteres Auswahlverfahren in die Förderung der Studienstiftung des deutschen Volkes aufgenommen. Außerdem werden Geld- und Sachpreise vergeben. Der Rechtsweg ist ausgeschlossen.

Zum Schluss noch einmal: Viel Spaß und viel Erfolg!

MCI-Sonderpreis

Beim Bundeswettbewerb Informatik spielt die Mensch-Computer-Interaktion und damit die Benutzungsschnittstelle eines eingesandten Programms für die Bewertung prinzipiell keine Rolle. Für die Benutzbarkeit von Informatiksystemen ist diese Komponente aber von ganz entscheidender Bedeutung, und so wird bei einigen Einsendungen zum BWINF erhebliche Mühe auf den Interaktionsaspekt verwendet.

Diese Mühe soll belohnt werden: Der Fachbereich Mensch-Computer-Interaktion (MCI) der Gesellschaft für Informatik e.V. (GI) schreibt erneut einen Sonderpreis für besonders gelungene Benutzungsschnittstellen aus. Verliehen wird dieser Preis auf der Tagung „Mensch & Computer 2011“, die vom 11. bis zum 14. September 2011 in Chemnitz stattfinden wird.

Für Bewerbungen um den MCI-Sonderpreis sollst du für eines deiner im Rahmen der Aufgabebearbeitung entwickelten Programme eine besonders innovative und gebrauchstaugliche Benutzungsschnittstelle entwickeln – auch wenn die zugehörige Aufgabe die Entwicklung einer Benutzungsschnittstelle nicht zwingend erfordert. Die zu entwickelnde Benutzungsschnittstelle kann also zusätzliche, für die Softwarenutzung sinnvolle Funktionen beinhalten, die für die Lösung der eigentlichen Aufgabe nicht erforderlich sind.

Beschreibe die Benutzungsschnittstelle in einem separaten Dokument. Gib aber nicht nur eine Bedienungsanleitung, sondern erläutere vor allem Entwurfskonzept und -entscheidungen. Bewertet wird deine Bewerbung danach, wie gut dein System folgende Kriterien erfüllt:

Dialogkriterien: Der Dialog eines Benutzers (oder einer Benutzerin) mit einem System ist *aufgabenangemessen*, wenn das System den Benutzer bei der Ausführung der Aufgaben unterstützt und sinnvoll führt, ohne Handlungen unnötig einzuschränken; *selbsterklärend*, wenn jeder Dialogschritt durch Reaktionen des Systems sofort verständlich wird oder dem Benutzer auf Anforderung erklärt wird; *kontrollierbar*, wenn der Benutzer in jeder Situation Richtung und Geschwindigkeit der Interaktion bestimmen kann, bis sein Ziel erreicht ist; *fehlertolerant*, wenn trotz offensichtlicher Fehler in der Eingabe das beabsichtigte Ergebnis ohne bzw. mit nur geringem Eingreifen des Benutzers erzielt werden kann.

Präsentationskriterien: Die Präsentation von Information ist *fachlich gut gestaltet*, wenn zusammengehörige Informationen räumlich gruppiert sind, ein „aufgeräumter“ Eindruck entsteht; *übersichtlich*, wenn die Menge der Informationen knapp und strukturiert dargestellt wird; *lesbar, klar und präzise*, wenn die (visuelle) Darstellung der Information leicht zu lesen bzw. erkennen ist und der Informationsgehalt ohne überflüssige Informationen vermittelt wird.

BWINF-Kriterien: Ein Programm und seine Benutzungsschnittstelle sind *originell*, wenn die Benutzungsschnittstelle ungewöhnlich und mit eigenen Mitteln (aber dennoch ergonomisch) gestaltet ist; *inspizierbar*, wenn die Benutzungsschnittstelle vollen Zugang zur Funktionalität erlaubt, also z. B. Kontrollflüsse dargestellt werden oder Parameter beeinflusst werden können.

Aufgabe 1: Kisten in Kisten in Kisten

Bea Y., 42, tauscht nach Angabe der Nachbarn häufiger große Teile ihres Hausrats übers Internet mit dem anderer Familien. Zum Versand braucht sie im Keller ein größeres Lager an quaderförmigen leeren Kisten. Jede Kiste lässt sich durch ein Tripel (a, b, c) von drei natürlichen Zahlen beschreiben, die ihre äußeren Maße, also Länge, Breite und Höhe, in Zentimetern angeben. Die inneren Ausmaße sind jeweils um einen Zentimeter kleiner.

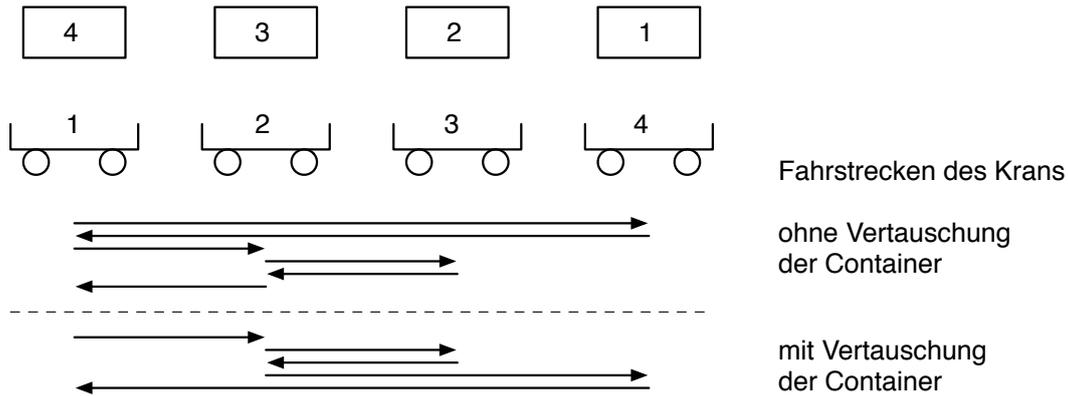
Da Frau Y. verständlicherweise unter Platzproblemen leidet, packt sie soweit möglich kleinere Kisten in größere. Aus der Schulzeit hat sie aber die Maxime „Binary is beautiful“ noch gut im Ohr, und so will sie nie in eine Kiste mehr als zwei andere Kisten direkt verpacken (während diese ihrerseits weitere Kisten enthalten dürfen). Der Einfachheit halber packt sie außerdem kleinere Kisten immer so in eine größere, dass jede Seite einer kleineren Kiste parallel zu einer Seite der größeren Kiste ist (keine „Schrägpackung“).

Aufgabe Hilf Frau Y., indem du ein Programm schreibst, das ihr benutzergerecht angeben kann, wie sie gegebene Kisten platzsparend wie beschrieben ineinander verschachteln kann. Dabei soll das Gesamtvolumen der äußeren (nicht in anderen Kisten enthaltenen) Kisten möglichst gering werden.

Aufgabe 2: Containerklamüsel

In einem Hafen werden Container aus einem Schiff geladen und einzeln auf Stellplätzen neben einem dort endenden Bahngleis abgestellt. Auf das Gleis fährt später ein Güterzug, der so zusammengestellt ist, dass nach dem Halt des Zuges je ein leerer Waggon einem Container gegenübersteht. Jetzt soll ein Kran, der zwischen Waggonen und Containern entlang fahren kann, die Container auf die einzelnen Waggonen laden. Beim Ausladen des Schiffes kann aber keine Rücksicht auf die Reihenfolge der Container genommen werden, so dass sie neben dem Gleis in einer beliebigen Reihenfolge stehen. Andererseits muss jeder Container auf einen bestimmten Waggon geladen werden, weswegen mit dem Kran auch die richtige Reihenfolge herbeigeführt werden muss.

Der Kran hat einen schwenkbaren Arm, der einen Container von seinem Stellplatz heben und ihn auf einen Waggon absetzen kann, eventuell nachdem der Kran ein Stück entlang des Gleises gefahren ist. Für die Benutzung des Krans fällt eine Gebühr an, die proportional zur Anzahl der insgesamt entlang des Gleises gefahrenen Strecke ist, während Hebe-, Absetz- und Schwenkbewegungen nicht erfasst werden und deswegen als gratis angesehen werden können. Die Betreiberfirma möchte den Kran so einsetzen, dass jeder Container auf den richtigen Waggon kommt und die dabei anfallende Gebühr so niedrig wie möglich ist.



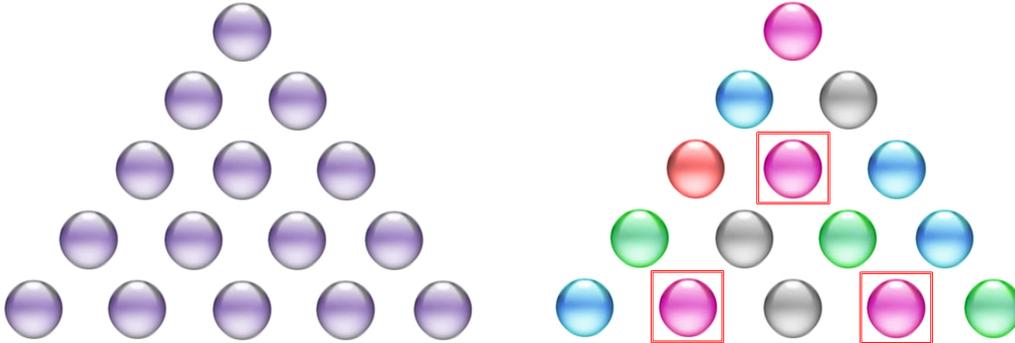
Nummerieren wir der Einfachheit halber die Containerstellplätze und die Waggonen $1, 2, \dots$ vom Ende des Bahngleises aus gerechnet, und sei Container i derjenige Container, der am Ende auf Waggon i sein soll, für $i = 1, 2, \dots$. Der Kran fängt immer an Platz 1 an und soll am Ende auch dorthin zurückgebracht werden. Im abgebildeten Beispiel ist Container i mit der Zahl i beschriftet, für $i = 1, \dots, 4$. In diesem Fall kann man so vorgehen, um die Container richtig auf die Waggonen zu bringen: Zuerst hebt der Kran Container 4 auf, fährt mit ihm zu Platz 4 und setzt ihn auf Waggon 4 ab. Danach wird in derselben Weise Container 1 auf Waggon 1 gebracht. Anschließend fährt der Kran ohne Last von Platz 1 nach Platz 2, die Container 3 und 2 werden ähnlich wie die Container 4 und 1 auf ihren jeweiligen Waggon gebracht, und zum Schluss fährt der Kran ohne Last von Platz 2 zu seiner vorgeschriebenen Endposition auf Platz 1. Die im Beispiel insgesamt gefahrene Strecke ist zweimal der Abstand zwischen Platz 1 und Platz 4 plus zweimal der Abstand zwischen Platz 1 und Platz 3, und das halten viele in der Firma für optimal.

Bis ein Praktikant (früherer BwInf-Teilnehmer) eines Tages versteht, warum der schwenkbare Kranarm sich um seine Mitte dreht und an beiden Enden eine Hebevorrichtung hat: Wenn der Kran an einem Platz i steht, können ein Container vom Stellplatz i und ein Container vom Waggon i gleichzeitig gehoben, geschwenkt und schließlich am jeweils anderen Ort abgesetzt werden; die beiden Container werden also vertauscht. Der Kran kann aber weiterhin maximal einen Container befördern, wenn er entlang des Gleises fährt. Jetzt kann die Beispielinstantz so gelöst werden, dass die insgesamt gefahrene Strecke zweimal der Abstand zwischen Platz 1 und Platz 4 plus zweimal der Abstand zwischen Platz 2 und Platz 3 ist, also kürzer als vorher. Die Lösung fängt damit an, Container 4 auf Waggon 2 und Container 3 auf Waggon 3 zu bringen. Danach wird Container 2 gehoben und zu Platz 2 gefahren, wo er mit dem auf Waggon 2 wartenden Container 4 vertauscht wird. Der Rest ist für den erfahrenen Kranführer ein Pappenstiel.

Aufgabe Nach der Entdeckung des Praktikanten muss neue Software zur möglichst optimalen Kransteuerung her. Der Praktikant ist aber längst wieder an seine Universität zurückgekehrt. Hilf der Firma!

Aufgabe 3: Traumdreiecke

Johanna hatte gestern Nacht einen merkwürdigen Traum. Über ihr schwebte ein großes Dreieck aus durchsichtigen Murmeln (im Bild links):



Wenn sie eine beliebige Murmel anfasste, dann konnte sie sich eine Farbe für diese wünschen, und die Veränderung der Farbe wurde durch eine schöne Melodie begleitet.

Als sie so alle Murmeln gefärbt hatte, wurde der Himmel dunkel, und eine tiefe Stimme verkündete: „Das ist keine schöne Färbung. Du darfst kein einziges gleichseitiges Dreieck aus zwei Murmeln in derselben Ebene und einer Murmel auf einer höheren Ebene so färben, dass alle seine Ecken die gleiche Farbe haben!“ Johanna betrachtete noch einmal ihre Färbung und entdeckte das Problem (im obigen Bild rechts).

Sie dachte „Pah, das ist doch einfach“, und färbte alle Murmeln mit unterschiedlichen Farben. Der Himmel wurde aber nicht klarer, und die Stimme ermahnte sie: „Das kann ja jeder! Aber mit wie vielen Farben kann man denn auskommen?“ Schweißgebadet wachte sie auf und setzte sich an ihren Rechner.

Aufgabe Schreibe ein Programm, das zu einer einzulesenden Zahl n zwischen 1 und 1000 eine Färbung des Dreiecks mit n Murmeln auf der untersten (ersten) Ebene, $n - 1$ Murmeln auf der zweiten Ebene usw. bis zu einer Murmel auf der n -ten Ebene ermittelt, die die Bedingung erfüllt, dass es kein gleichseitiges Dreieck gibt, dessen Ecken zwei Murmeln in derselben Ebene und eine Murmel auf einer höheren Ebene sind, welche alle die gleiche Farbe haben. Dabei sollen so wenige Farben wie möglich verwendet werden. Die Färbung und die Anzahl verwendeter Farben sollen ausgegeben werden. Das Format der Ausgabe der Färbung sieht für das Beispiel-Dreieck so aus:

```
1
2 3
4 1 2
5 3 5 2
2 1 3 1 5
```