

34. Bundeswettbewerb Informatik 2015/2016

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;
- 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 eine Einladung zu den „Forschungstagen Informatik 2016“ des Max-Planck-Instituts für Informatik in Saarbrücken.

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

Es gibt drei Aufgaben. **Eine Einsendung darf 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.

An dieser Runde dürfen nur Einzelpersonen teilnehmen, die in der ersten Runde in drei Aufgaben insgesamt mindestens 12 Punkte erreicht oder einem Team angehört haben, dem dieses gelungen ist. Gruppenarbeit ist in der zweiten Runde nicht zulässig.

Einsendeschluss ist der 11. April 2016.

Bearbeitung

Die Bearbeitung einer Aufgabe sollte zunächst eine 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.

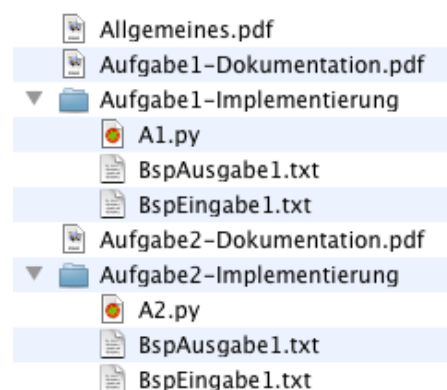
Grundsätzlich gelten die Vorgaben der 1. Runde weiter. Wesentliches Ergebnis der Aufgabebearbeitung ist also eine **Dokumentation**, in der du den *Lösungsweg* sowie die *Umsetzung* des Lösungswegs in das dazugehörige Programm beschreibst. Die Beschreibung des Lösungswegs kann mit Hilfe (halb-)formaler Notationen präzisiert werden, die Beschreibung der Umsetzung mit Verweisen auf die entsprechenden Quellcode-Elemente. In die Dokumentation gehören auch *Beispiele* (Programmein- und -ausgaben oder Zwischenschritte), die zeigen, wie das Programm sich in unterschiedlichen Situationen verhält. Komplettiert wird die Dokumentation durch *Auszüge aus dem Quelltext*, die alle wichtigen Module, Methoden, Funktionen usw. enthalten.

Das zweite Ergebnis der Bearbeitung ist die **Implementierung**. Sie besteht aus dem zur Lösung der Aufgabe geschriebenen lauffähigen *Programm* und dem vollständigen *Quelltext*. Außerdem können Beispielein- und -ausgaben oder weiteres hilfreiches Material der Implementierung beigefügt werden.

Die Dokumentation zu einer Aufgabe mit allen oben genannten Bestandteilen muss als ein PDF-Dokument eingereicht werden. Dieses Dokument wird für die Bewertung ausgedruckt. **Es kann sein, dass für die Bewertung nur die Dokumentation herangezogen wird.** Diese sollte also einen lückenlosen und nachvollziehbaren Nachweis des Leistungsumfangs und der Funktionstüchtigkeit der Programme geben – und unbedingt auch Beispiele enthalten! Der Umfang der Dokumentation soll sich in Grenzen halten; eine gute Dokumentation vermittelt kurz und präzise alles Nötige, insbesondere die wesentlichen Ideen beim Lösungsweg. Nötig ist alles, was Interessierte mit guten Informatikkenntnissen, die die Aufgabenstellung kennen, wissen müssen, um den Lösungsweg zu verstehen und seine Umsetzung nachzuvollziehen. Entscheidend für eine gute Bewertung sind zwar richtige (und sauber umgesetzte) Lösungswege, aber die Dokumentation hat schon oft den Ausschlag für oder gegen das Weiterkommen gegeben.

Einsendung

Die Einsendung erfolgt wieder über das BWINF-PMS (pms.bwinf.de). Hochladen kannst du ein max. 20 MB großes ZIP-Archiv (z.B. VornameNachname.zip); seine Inhalte sollten so strukturiert sein wie im Bild rechts. Ein Dokument `Allgemeines.pdf` ist aber nur nötig, wenn du allgemeine, von den Aufgabebearbeitungen unabhängige Anmerkungen zu deiner Einsendung machen willst. Die Dokumentationen müssen als PDF-Dokumente enthalten sein. Dateien in anderen Formaten werden möglicherweise ignoriert. Die Schriftgröße einer Dokumentation muss mindestens 10 Punkt sein, bei Quelltext mindestens 8 Punkt. Auf jeder Seite einer Dokumentation sollen in der Kopfzeile Verwaltungsnummer, Vorname, Name und Seitennummer stehen. Die Verwaltungsnummer steht auf der Teilnahmebescheinigung der ersten Runde.



Weitere Hinweise

Bei der Bewertung können Programme unter Windows (7 / 8 / 10), Linux, Mac OS X (10.11) und Android ausgeführt werden.

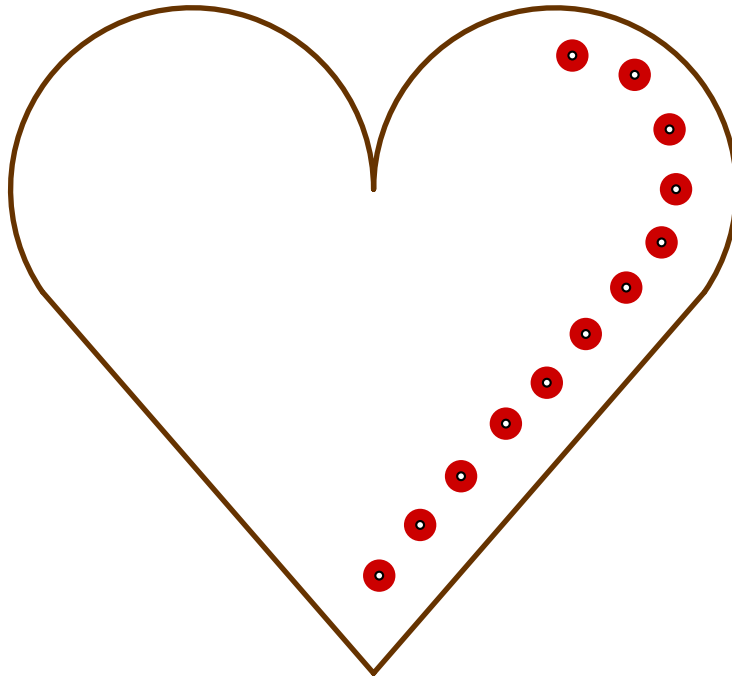
Fragen zu den Aufgaben können per Mail an kontakt@bundeswettbewerb-informatik.de oder telefonisch unter 0228-378646 (zu üblichen Arbeitszeiten) gestellt werden. Die Antwort auf E-Mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf unseren Webseiten (www.bundeswettbewerb-informatik.de).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

Allen Teilnehmern der zweiten Runde wird bis Mitte Juni 2016 die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die im September 2016 ausgerichtet 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!

Aufgabe 1: Geburtstagskuchen

Linda hat für ihre Schwester einen Geburtstagskuchen in Herzform gebacken und Kerzen dafür gekauft. In einem Moment der Unachtsamkeit hat ihr kleiner Bruder Luis die Kerzen bereits auf den Kuchen gesteckt:



Doch diese Verteilung der Kerzen gefällt Linda gar nicht; sie möchte die Kerzen möglichst gleichmäßig auf dem ganzen Kuchen verteilen.

Aufgabe

1. Definiere ein formales Maß für die Gleichmäßigkeit einer Verteilung von Kerzen. Schreibe ein Programm, das die Tortenfläche sowie eine Liste von Koordinaten der einzelnen Kerzen einliest und den Wert deines Maßes ausgibt.
2. Schreibe ein Programm, das auf einer herzförmigen Tortenfläche eine gegebene Anzahl von Kerzen verteilt und ein Bild der Verteilung ausgibt. Sorge dafür, dass die Verteilung nach deinem Maß schön gleichmäßig und für Lindas Kuchen auf jeden Fall besser ist als die von Luis. Wende dein Programm auf verschiedene Kerzenzahlen an.

Aufgabe 2: Mißglückte Drohnenlieferung

In der zukunftsorientierten Mustersiedlung Amacity stehen 100 Häuser in einem Quadrat angeordnet. Da Pakete in Zukunft vielleicht durch kleine, fliegende Zustellroboter geliefert werden, ist ihr Dach schon mit einer geeigneten Empfangsplattform ausgestattet, auf welcher aber nur Platz für ein großes Paket ist.

In einem einmaligen Experiment werden jetzt tatsächlich 100 Pakete an diese 100 Häuser geliefert und alle heil auf die Empfangsplattformen abgelegt. Leider wurde eine falsche Adressenliste verwendet und die Pakete sind nicht an die richtigen Häuser geliefert worden. Die folgende Zeichnung zeigt, wo die Pakete gelandet sind. Leider muss aber das Paket i, j tatsächlich in die i -te Zeile und in die j -te Spalte (wobei die Nummerierung mit 0 beginnt und mit 9 endet).

4,0	8,7	3,4	3,3	7,5	4,3	6,5	3,5	5,7	2,7
0,5	7,0	2,3	8,8	0,0	4,8	0,8	1,3	7,3	0,1
7,9	3,0	9,4	6,1	8,0	6,2	0,4	3,9	2,8	3,8
1,6	8,5	4,6	5,8	6,6	1,7	4,1	6,0	5,2	9,9
7,2	2,9	4,2	1,1	9,8	5,0	5,3	6,7	9,7	8,4
1,2	4,5	5,6	9,5	6,9	1,9	8,1	9,6	2,1	0,7
0,9	6,8	4,9	6,4	8,2	3,1	9,2	4,4	7,7	9,1
5,5	1,5	2,6	5,4	8,3	1,4	3,7	1,8	7,4	7,1
3,2	1,0	9,3	4,7	0,3	7,8	8,6	8,9	2,4	3,6
7,6	2,5	0,2	5,1	5,9	6,3	9,0	2,0	2,2	0,6

Die Einwohner möchten jetzt die Pakete schrittweise an die richtigen Orte befördern. Dafür steigt aus jedem Haus eine Person auf das Dach.

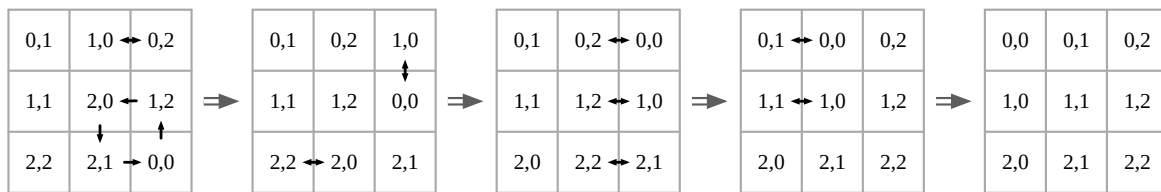
Ein Schritt sieht dann so aus:

- Die Person auf dem Dach nimmt das dort liegende Paket in die Hand.
- Sie sieht auf einem Plan nach, ob und gegebenenfalls in welche Richtung das Paket in diesem Schritt geworfen werden soll.
- Das Paket wird in diese Richtung geworfen und von der Person auf dem entsprechenden Nachbarhaus aufgefangen und dann dort abgelegt.

Wichtig dabei ist natürlich, dass in (c) alle Personen, die ein Paket werfen, dies gleichzeitig tun. In einem Schritt können also bis zu 100 Pakete gleichzeitig zu einem Nachbarhaus wechseln. Nachbarhäuser sind direkt, aber nicht diagonal aneinander grenzende Häuser.

Noch wichtiger aber ist, dass der Plan garantieren muss, dass nach jedem Schritt auf jedem Haus genau ein Paket liegt.

Zur Illustration zeigen wir einen vierschriftigen Plan für eine einfachere Siedlung mit nur neun Häusern:



Aufgabe

Erstelle mit Computerhilfe einen solchen Plan für die oben angegebene Paketverteilung in Amacity. Er soll übersichtlich zeigen, wer in jedem Schritt sein Paket in welche Richtung wirft. Versuche die Anzahl der Schritte klein zu halten. Auf der BWINF-Webseite findest du bereits einen Plan. Schaffst du es mit ähnlich vielen Schritten auszukommen? Oder kannst du sogar einen Plan mit weniger Schritten finden?

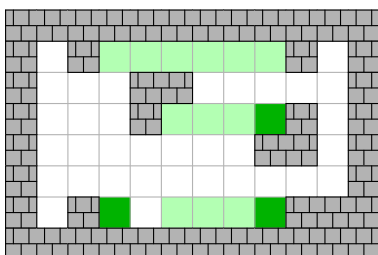
Gib deinen Plan auch in eine Datei aus, in dem auf der Webseite spezifizierten Format.

Es ist hübsch und sehr nützlich, alle Schritte zu visualisieren – am besten in einer Animation.

Aufgabe 3: Torkelnde Yamyams

In einer aus quadratischen Feldern bestehenden Welt bewegt sich ein Yamyam auf folgende Weise: Hat er sich einmal zufällig für eine der vier Himmelsrichtungen entschieden, so fährt er in dieser Richtung entlang, bis er auf ein Hindernis stößt. Dann setzt er seine Fahrt in eine zufällige freie Himmelsrichtung fort. Es gibt in dieser Welt *Ausgänge*. Es ist wünschenswert, dass ein Yamyam auf seiner zufallsgesteuerten Reise sicher zu einem Ausgang findet.

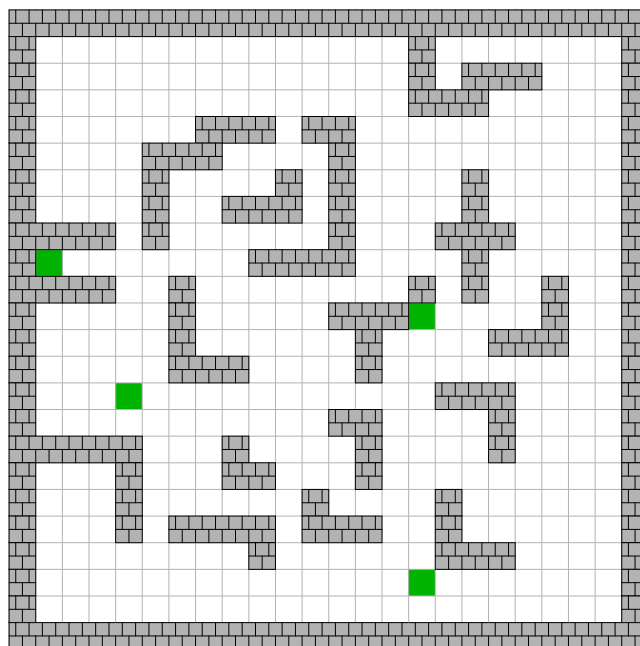
Wir nennen ein Feld *sicher*, wenn ein Yamyam von diesem aus irgendwann einen Ausgang erreicht, egal wie die zufälligen Richtungsänderungen ausfallen und wie lange die Fahrt dauert. Mathematisch ausgedrückt ist ein Feld sicher, wenn ein Yamyam von dort mit Wahrscheinlichkeit 1 einen Ausgang erreicht.



Im linken Beispiel sind die sicheren Felder hellgrün und die Ausgänge dunkelgrün eingezeichnet. Das oberste, linke hellgrüne Feld ist beispielsweise sicher: Ein Yamyam könnte zwar zunächst nach rechts und dann viele Male abwechselnd nach links und nach rechts fahren. Aber irgendwann wird er auch einmal den Weg nach unten wählen, wenn er an eine der beiden Wände stößt, und dann direkt einen Ausgang erreichen.

Aufgabe

Schreibe ein Programm, das eine Beschreibung solch einer Welt einlesen kann und dann die sicheren Felder bestimmt. Lasse dein Programm auf den Beispielen laufen, die du auf der BWINF-Webseite findest. Diese Beispiele enthalten auch folgende Welt:



Bei dieser Welt ist es wahrscheinlich zu mühsam, die sicheren Felder ohne Computerhilfe zu bestimmen.