

Die Aufgaben der 2. Runde

Allgemeine Hinweise

Herzlichen Glückwunsch zum Erreichen der 2. Runde! Hier sind die Aufgaben. Sie sind anspruchsvoll, und ihre Bearbeitung ist aufwändig. Aber die Mühe lohnt sich, denn durch Teilnahme an der 2. Runde

- wirst du sicher sehr viel lernen;
- kannst du dich für die Endrunde qualifizieren;
- kannst du einen Buchpreis der Verlage O'Reilly oder dpunkt.verlag gewinnen;
- hast du am Ende eine Arbeit fertig gestellt, die du als 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 2020“ 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 1. Runde in drei Aufgaben insgesamt mindestens 12 Punkte erreicht oder einem Team angehört haben, dem dieses gelungen ist. Gruppenarbeit ist in der 2. Runde nicht zulässig.

Einsendeschluss ist Montag, der 20. April 2020.

Bearbeitung

Die Bearbeitung einer Aufgabe sollte zunächst eine nachvollziehbare und vollständige Lösung aller Teilaufgaben enthalten. **Zusatzpunkte** 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, die praktisch realisiert werden; 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 darstellte.

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 aussagekräftige *Beispiele* (Programmeingaben/-ausgaben inklusive ggf. Zwischenschritte/-ergebnisse), die zeigen, wie das Programm sich in unterschiedlichen Situationen verhält. Komplettiert wird die Dokumentation am Ende durch *Auszüge aus dem Quelltext*, die alle wichtigen Module, Methoden, Funktionen usw. enthalten. Im Text des Lösungswegs und der Umsetzung sollte jedoch kein oder nur wenig Quellcode enthalten sein.

Weiteres Ergebnis der Aufgabenbearbeitung ist die **Implementierung**. Sie besteht aus dem zur Lösung der Aufgabe geschriebenen lauffähigen *Programm* und dem vollständigen *Quelltext*. Außerdem können Beispieleingabe/-ausgaben oder weiteres hilfreiches Material der Implementierung beigelegt werden.

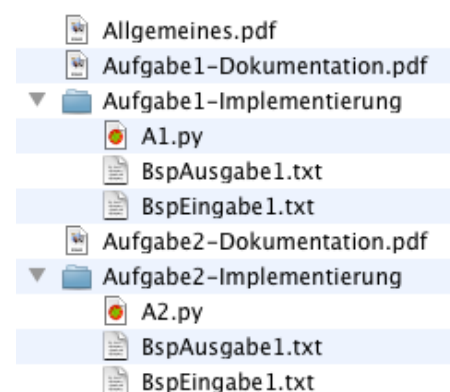
Die Dokumentation zu einer Aufgabe mit allen oben genannten Bestandteilen muss als PDF-Dokument eingereicht werden. Dieses Dokument wird für die Bewertung ausgedruckt. **Es kann sein, dass für die Bewertung deiner Einsendung nur die Dokumentation herangezogen wird.** Sie sollte also einen lückenlosen und verständlichen Nachweis des Leistungsumfangs und der Funktionstüchtigkeit der Programme geben – und unbedingt die vorgegebenen Beispiele neben eigenen 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. Das Erstellen der Dokumentation sollte die Arbeit an Lösungsideen und Umsetzung eng begleiten. Wer zunächst die Lösungsidee verständlich formuliert, dem fällt anschließend eine fehlerlose Implementierung leichter. Abbildungen tragen in der Regel zur Verständlichkeit bei, und es schadet nicht, die Dokumentation von Dritten prüfen zu lassen, selbst wenn sie fachfremd sind.

Einsendung

Die Einsendung erfolgt wieder über das BWINF-PMS (pms.bwinf.de). Hochladen kannst du ein max. 40 MB großes ZIP-Archiv (z. B. `VornameNachname.zip`); sein Inhalt sollte so strukturiert sein wie rechts abgebildet. Die Dokumentationen der bearbeiteten Aufgaben müssen als PDF-Dokumente enthalten sein; Dateien in anderen Formaten werden möglicherweise ignoriert. Ein Dokument `Allgemeines.pdf` ist nur dann nötig, wenn du allgemeine, von den Aufgabenbearbeitungen unabhängige Bemerkungen zu deiner Einsendung machen willst. Die Schriftgröße einer Dokumentation muss mindestens 10 Punkt sein, bei Quelltext mindestens 8 Punkt. Auf jeder Seite einer Dokumentation sollen in der Kopfzeile die Teilnahme-Id, Vorname, Name und Seitennummer stehen; hierfür sind auf der BWINF-Webseite entsprechende Dateien als Vorlage. Die Teilnahme-Id steht auf der Teilnahmebescheinigung der 1. Runde.



Weitere Hinweise

Bei der Bewertung können Programme unter Windows (7 / 10), Linux, Mac OS X (10.14) und Android ausgeführt werden.

Fragen zu den Aufgaben können per Mail an bundeswettbewerb@bwinf.de oder, zu üblichen Arbeitszeiten, telefonisch unter 0228 378646 gestellt werden. Die Antwort auf E-Mail-Anfragen kann sich leicht verzögern. Informationen zur 2. Runde finden sich auf unseren Webseiten: bwinf.de/bundeswettbewerb. Unter einstieg-informatik.de findest du unsere Community; dort werden im Forum sicher wieder viele Teilnehmerinnen und Teilnehmer über die Aufgaben diskutieren – ohne Lösungsideen auszutauschen.

Allen Teilnehmern der 2. Runde wird bis Mitte Juni 2020 die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die 15.–18. September 2020 vom Department Informatik der Universität Erlangen-Nürnberg 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 Geldpreise vergeben. Der Rechtsweg ist ausgeschlossen.

Viel Spaß und viel Erfolg!

Aufgabe 1: Stromrallye

Stromrallye ist ein neues Knobelspiel:

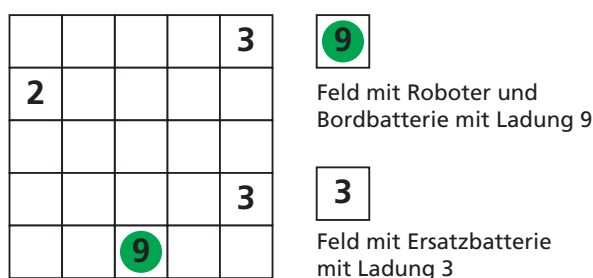
Auf einem quadratischen Spielbrett, das in quadratische Felder eingeteilt ist, bewegt sich ein Roboter. Der Roboter hat eine austauschbare Bordbatterie mit vorgegebener Ladung. Auf einigen Feldern liegen außerdem Ersatzbatterien mit unterschiedlichen Ladungen. Die Ladungen aller Batterien sind durch ganze Zahlen gegeben.

Der Roboter kann Schritte um ein Feld nach rechts, links, oben oder unten machen. Bei jedem Schritt sinkt die Ladung seiner Batterie um 1. Sinkt die Ladung auf 0, kann er keine weiteren Schritte machen.

Wenn der Roboter auf ein Feld mit einer Ersatzbatterie kommt, muss er seine Bordbatterie gegen die Ersatzbatterie austauschen, selbst wenn die Ersatzbatterie weniger Ladung hat als die Bordbatterie. Die Ersatzbatterie ist danach die neue Bordbatterie. Nach dem Tausch muss er das Feld verlassen.

Ziel des Spiels Stromrallye ist es, den Roboter so über das Spielbrett zu bewegen, dass am Ende alle Batterien, einschließlich der Bordbatterie, entladen sind.

Hier ist ein Beispiel für eine Spielsituation auf einem Spielbrett mit 5×5 Feldern:



Aufgabe

- a) Schreibe ein Programm, das Stromrallye spielen kann. Es soll zuerst eine Spielsituation einlesen: die Größe des Spielbretts, die Position des Roboters, die Ladung der Bordbatterie sowie die Positionen und Ladungen der Ersatzbatterien. Danach soll dein Programm bestimmen, ob die eingelesene Spielsituation lösbar ist, also ob der Roboter sich so bewegen kann, dass am Ende alle Batterien entladen sind. Wenn ja, soll dein Programm eine entsprechende Folge von Schritten ausgeben.
- b) Stromrallye soll auch von Menschen gespielt werden. Schreibe ein Programm, das Spielsituationen erzeugt, die lösbar, aber für einen menschlichen Spieler schwierig zu lösen sind. Dokumentiere, wie du dein Schwierigkeitsmaß definierst.

Hinweis: Zum Ausprobieren kannst du Stromrallye mit einem Damebrett nachspielen. Als Roboter nimmst du einen schwarzen Stein, eine Batterie mit einer Ladung von n ist ein Stapel aus $n + 1$ weißen Steinen. Die Bordbatterie wird auf den Roboterstein gestapelt.

Aufgabe 2: Geburtstag

2020 ist ein großes Jahr: Der Bundeswettbewerb Informatik feiert seinen 40. Geburtstag! Der Biber will das vorbereiten und die Jahreszahl möglichst elegant darstellen. Er hat sich ausgedacht, einen Term auszuknobeln, dessen Wert diese Zahl ist. Damit es aber wirklich eine Knobelei wird, soll der Term folgende Bedingungen erfüllen:

- Er verwendet nur eine vorgegebene Ziffer von 1 bis 9. Aus dieser Ziffer können auch mehrstellige Zahlen gebildet werden. Ist 4 die Ziffer, sind also auch 44, 444, 4444 und so weiter erlaubt.
- Er darf die Grundrechenarten +, −, *, / enthalten.
- Er darf beliebige Klammern enthalten, wenn sie korrekt gesetzt sind.
- Der Term muss mit so wenigen Ziffern wie möglich auskommen.

Der Biber hat das für das aktuelle Jahr 2019 schon einmal für jede Ziffer ausgeknobelt. Er kommt zum Beispiel auf

$$\begin{aligned} &(((1+1) * ((11111-1) / 11)) - 1) \\ &((2 * ((22 * (2 + (2 * 22))) - 2)) - (2 / 2)) \\ &(3 + ((3+3) * (3+333))) \\ &(4 + (((4+4) * ((4 * (4 * (4 * 4))) - 4)) - (4 / 4))) \\ &((5 * (5 + (5 * ((5 * 5) + 55)))) - (5 + (5 / 5))) \\ &((6 + (6 * (6 + 666))) / ((6 + 6) / 6)) \\ &(((77 - 7) / 7) - (7 * (7 - (7 * ((7 * 7) - 7)))))) \\ &(88 + ((88 / 8) - ((8 + 8) * (8 - (8 * (8 + 8)))))) \\ &(9 + ((9 + (99 + ((9 + 9) * 999))) / 9)) \end{aligned}$$

Aufgabe

- Schreibe ein Programm, das zu einer eingegebenen natürlichen Zahl und einer Ziffer zwischen 1 und 9 einen Term ausgibt, der den genannten Bedingungen des Bibers genügt. Welche Terme findet dein Programm für das Jahr 2020 und die ebenfalls interessanten Jubiläumsjahre 2030, 2080 und 2980?
- Wenn man zusätzlich zu den Grundrechenarten die Verwendung von Potenz- und der Fakultätsfunktion erlaubt, kann man in einigen Fällen mit noch weniger Ziffern auskommen. Zum Beispiel ist $2019 = (((3!)!) + (3 + ((3!) * ((3!)^3)))$. Während der Term ohne Fakultät und Potenzen siebenmal die Ziffer 3 verwendet, kommt dieser Term mit nur fünf Dreien aus.

Erweitere dein Programm so, dass es versucht, für die gegebene Zahl einen so erweiterten Term zu finden, der noch weniger Ziffern als die Lösung in (a) enthält. Beschreibe deine dabei verwendete Strategie.

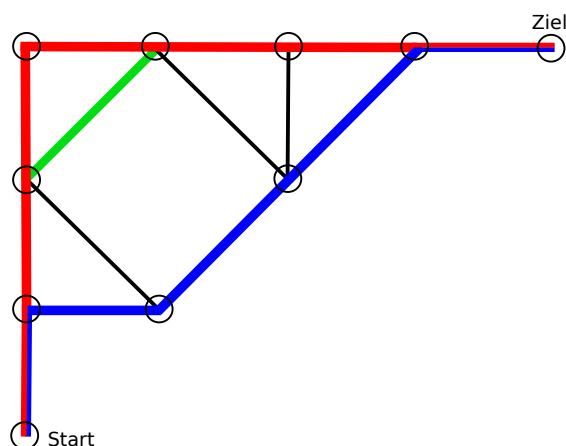
Aufgabe 3: Abbiegen?

Bilal möchte mit seinem Fahrrad zu einem Freund fahren. Der kürzeste Weg führt über viele kleine Straßen und ist verwinkelt. Bilal fährt an Kreuzungen am liebsten geradeaus und ist daher bereit, einen etwas längeren Weg in Kauf zu nehmen, wenn er dafür weniger oft abbiegen muss.

Am unten abgebildeten Beispielwegenetz sieht man, dass weniger Abbiegen tatsächlich zu einer längeren Wegstrecke führen kann. Der kürzeste, blaue Weg hat die Länge $3 + 2 \cdot \sqrt{2} \approx 5,83$ und erfordert dreimaliges Abbiegen. Der rote Weg mit einmaligem Abbiegen und der Länge 7 ist demgegenüber um etwa 20 % länger.

Bei gutem Wetter macht es Bilal nichts aus, einen um 30 % längeren Weg zu fahren. Bei schlechtem Wetter sollte die Verlängerung allerdings maximal 15 % betragen. Bei Sonnenschein ist beim gegebenen Beispiel also der längere rote Weg akzeptabel. Bei Regen ist der rote Weg zu lang, und Bilal ist dann bislang den blauen Weg gefahren.

Neulich hat Bilal aber entdeckt, dass er den roten Weg abkürzen kann, indem er die grüne Straße nimmt. Dieser rot-grüne Weg ist $5 + \sqrt{2} \approx 6,41$ lang und damit um etwa 10 % länger als der kürzeste Weg. Bei schlechtem Wetter fährt Bilal also nun den rot-grünen Weg, auf dem er einmal weniger abbiegen muss als auf dem blauen Weg.



Aufgabe

Hilf Bilal und schreibe ein Programm, das eine Straßenkarte mit Start- und Zielpunkt einliest und ihm einen Weg vorschlägt. Damit das Programm bei jedem Wetter nutzbar ist, soll der Benutzer die maximale Verlängerung in Prozent eingeben können.