

## 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 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 2018“ 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 9. April 2018.**

## 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*

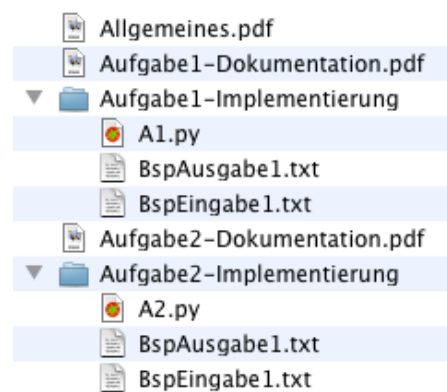
des Lösungswegs in das dazugehörige Programm beschreibst. Die Beschreibung des Lösungswegs kann mit Hilfe (halb-)formaler Notationen präzisiert werden, die Beschreibung der Umsetzung mit Verweisen auf die entsprechenden Quellcode-Elemente. In die Dokumentation gehören auch *Beispiele* (Programmein- und -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 Beispielin- 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](http://pms.bwinf.de)). Hochladen kannst du ein max. 40 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 Aufgabenbearbeitungen 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 (10.12) und Android ausgeführt werden.

**Fragen zu den Aufgaben** können per Mail an [kontakt@bundeswettbewerb-informatik.de](mailto: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](http://www.bundeswettbewerb-informatik.de)). In der Community von [einstieg-informatik.de](http://einstieg-informatik.de) werden sicher wieder viele Teilnehmer über die Aufgaben diskutieren – ohne Lösungsideen auszutauschen.

Allen Teilnehmern der zweiten Runde wird bis Mitte Juni 2018 die Bewertung mitgeteilt. Die Besten werden zur Endrunde eingeladen, die im September 2018 von der Universität Paderborn 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.

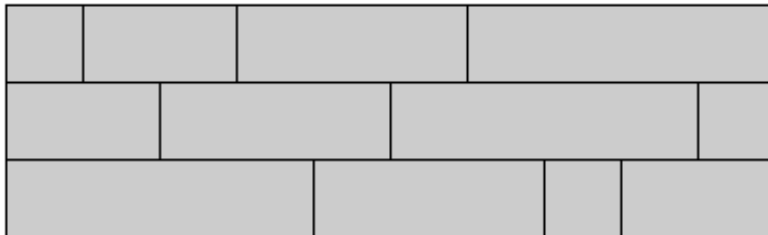
Zum Schluss noch einmal: Viel Spaß und viel Erfolg!

## Aufgabe 1: Die Kunst der Fuge

Ilona besitzt einen riesigen Haufen Holzklötzchen: Diese haben alle dieselbe Höhe und Tiefe, aber verschiedene Längen.

Ilona möchte eine Mauer bauen. Jede Reihe der Mauer soll aus  $n$  Klötzchen bestehen, die die Längen 1 bis  $n$  haben und lückenlos aneinander liegen. Die Stellen zwischen den Klötzchen heißen Fugen. Ilona möchte, dass in der fertigen Mauer niemals zwei Fugen übereinander liegen, selbst wenn sich mehrere Reihen dazwischen befinden. Außerdem soll ihre Mauer möglichst hoch sein.

Für  $n = 4$  gelingt es ihr recht schnell, eine Mauer mit drei Reihen zu bauen:

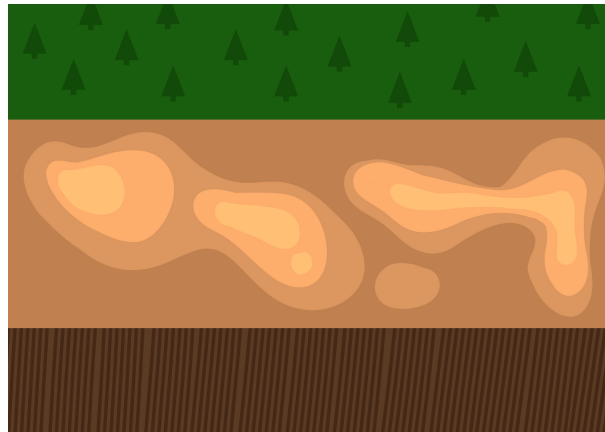


### Aufgabe

Hilf Ilona, indem du ein Programm schreibst, das nach Eingabe von  $n$  eine nach ihren Vorgaben konstruierte, möglichst hohe Mauer ausgibt. Für  $n = 10$  sollte dein Programm eine Mauer der Höhe 6 ausgeben können. Wie hoch werden die Mauern deines Programms für größere  $n$ ?

## Aufgabe 2: Wehret den Wildschweinen!

Ein Wald ist von Wildschweinen bevölkert. Sie fallen regelmäßig auf einem südlich des Waldes liegenden Acker ein. Zwischen Wald und Acker liegt ein rechteckiges, hügeliges Brachland, das im Osten und im Westen abgezaunt ist.



Der Besitzer des gesamten Geländes ist es leid, dass die Wildschweine ständig seinen Acker verwüsten. Er will dafür sorgen, dass sie nicht vom Wald auf den Acker gelangen können. Dafür macht er sich zunutze, dass Wildschweine nicht in der Lage sind, größere Steigungen zu überwinden. Er kann die Wildschweine also davon abhalten, das Brachland zu überqueren, indem er an gezielten Stellen dessen Landschaft so verändert, dass sie für ein Wildschwein zu steil abfällt oder ansteigt.

Zur Vereinfachung reduziert er die Landschaft auf Planquadrate und die möglichen Wege der Wildschweine auf solche, deren Abschnitte stets vom Zentrum eines Planquadrats zum Zentrum eines in nördlicher, südlicher, östlicher oder westlicher Richtung benachbarten Planquadrats verlaufen. Er muss jetzt erreichen, dass auf jedem solchen Weg von der Nordgrenze bis zur Südgrenze der Brache mindestens einmal der Höhenunterschied zwischen aufeinander folgenden Planquadraten mindestens 1 m beträgt. Dazu kann er Erde von einem Planquadrat zu einem benachbarten Planquadrat bringen lassen, wodurch die Höhe des ersten Planquadrats fällt und die des zweiten Planquadrats um genauso viel wächst. Da solche Erdarbeiten aber teuer sind, ist er bestrebt, diese Maßnahme so wenig wie möglich einzusetzen.

### Aufgabe

Schreibe ein Programm, das eine Matrix mit den Höhen der Planquadrate des Brachlands einliest und kostengünstigste Erdarbeiten vorschlägt, die jeden Weg von der Nordgrenze bis zur Südgrenze für Wildschweine unpassierbar werden lassen. Gehe dabei davon aus, dass es genau  $h$  Euro kostet,  $h$  cm Erde von einem Planquadrat zu einem benachbarten Planquadrat bringen zu lassen.

Wende dein Programm auf die Beispiele an, die du auf der BWINF-Website findest, und dokumentiere die Ergebnisse.

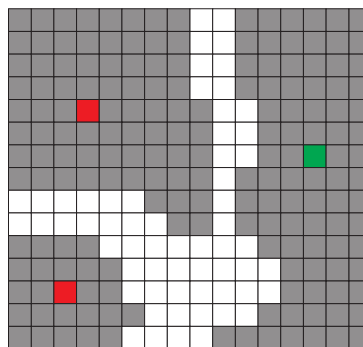
### Aufgabe 3: Quo vadis, Quax?

In einer abgelegenen und unwegsamen Gegend will Quax die Fähigkeiten seines neuen Quadcopters ausprobieren. Da wird aus dem Spiel Ernst: Ein Tornado mit anschließender Überschwemmung vernichtet seine Essenvorräte, so dass er seinen Urlaub abbrechen und sofort die lange Wanderung zurück zur einzigen Stadt der Provinz antreten muss. Außerdem hat sich die Landschaft vollkommen verändert – es haben sich Flüsse und Seen gebildet, wo früher keine waren. Es ist nicht klar, ob es überhaupt noch einen gangbaren Weg zurück zur Stadt gibt und, falls ja, wo sich ein solcher Weg zwischen den Gewässern durchschlängelt.

Quax möchte die Landschaft mit dem Quadcopter erkunden, bevor er loslegt, um lange Umwege zu vermeiden, die er sich nicht leisten kann. Der Quadcopter kann in einer frei wählbaren Flughöhe einen beliebigen Punkt der Landschaft anfliegen. Allerdings ist auch er beschädigt: Die Speicherkarte ist hin, und es funktioniert nur ein einziger Sensor. Dieser erfasst ein quadratisches, nach Norden ausgerichtetes Gebiet unter dem Quadcopter, dessen Seitenlänge proportional zur Flughöhe ist. Der Sensor kann für dieses Testgebiet entscheiden, ob es ganz von Wasser bedeckt ist („Wasser“), ganz aus trockenem Land besteht („Land“) oder sowohl Wasser als auch Land enthält („Gemischt“).

Quax kann wiederholt den Quadcopter für eine bestimmte Kombination aus Flughöhe und Zielpunkt programmieren, ihn entsenden und bei seiner Rückkehr das Ergebnis („Wasser“, „Land“ oder „Gemischt“) in Empfang nehmen. Liefert der Quadcopter auch für ein kleinstmögliches Testgebiet mit Seitenlänge 20 m immer noch das Ergebnis „Gemischt“, geht Quax davon aus, dass er dieses Testgebiet überqueren kann – zur Not schwimmt er ein paar Meter mit dem Gepäck.

Das folgende Pixelbild zeigt eine Beispiel-Landschaft. Jedes Pixel repräsentiert einen quadratischen Bereich der Landschaft mit Seitenlänge 10 m. Das grüne Pixel zeigt die Position der Stadt. Vom oberen roten Startpunkt kann Quax die Stadt erreichen, vom unteren aber nicht.



### Aufgabe

- Schreibe ein Programmmodul, das aus den Ergebnissen durchgeführter Flüge bestimmt, ob es für Quax einen gangbaren Weg zurück zur Stadt gibt, ob es definitiv keinen solchen Weg gibt oder ob mehr Flüge nötig sind, um eine Entscheidung herbeizuführen.
- Beschreibe eine Strategie für das Entsenden des Quadcopters, die mit möglichst wenig Flügen einen Weg zur Stadt findet, falls es einen gibt.

- (c) Schreibe ein Programm, das eine Landschaft einliest, in welcher die Stadt sowie mehrere mögliche Standorte von Quax markiert sind. Ein Pixel repräsentiert ein Quadrat der Seitenlänge 10 m. Dann soll das Programm die anhand der Strategie aus (b) festgelegten Entsendungen – für jeden Standort des Quadcopters einzeln – simulieren. Ausgeben soll es je eine Darstellung der Landschaft, in der die Testgebiete mit ihren jeweiligen Ergebnissen geeignet eingetragen sind. Falls es einen gangbaren Weg zur Stadt gibt, soll dieser in der Darstellung erkennbar sein.

Wende dein Programm mindestens auf die auf der BWINF-Website gegebenen Beispiele an und dokumentiere die Ergebnisse. Gehe davon aus, dass sich außerhalb der eingelesenen Landschaft nur Wasser befindet.

- (d) Gibt es Verteilungen von Flüssen und Seen, bei denen deine Strategie aus (b) zu sehr vielen Flügen führt? Falls nein, erkläre warum nicht. Falls ja, gib ein Beispiel dafür an und erkläre, worin das Problem besteht.