



---

## Allgemeine Informationen und Regeln

Der theoretische Teil der zweiten Runde findet am 28.2.2015 an der ETH Zürich statt. Die Teilnehmer müssen in vier Stunden vier Aufgaben lösen. Die Lösungen müssen auf Papier geschrieben werden. Verschiedene Aufgaben sollten auf verschiedene Blätter geschrieben werden. Den Teilnehmenden ist es nicht erlaubt, Literatur (z.B. Bücher, Listen von Programmen etc.), elektronische Geräte (ausser einfachen Uhren) oder andere Hilfsmittel zu gebrauchen.

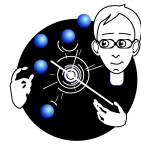
Die Lösungen werden nach ähnlichen Kriterien wie die theoretischen Aufgaben der ersten Runde bewertet. Die wichtigsten Kriterien sind die Korrektheit und die asymptotische Laufzeit der Lösung. Die Qualität der Beschreibung und die Argumente für die Korrektheit der Lösung werden ebenfalls bewertet.

Die Lösung sollte enthalten:

- **Beschreibung der Lösung.** Dieser Teil sollte den benutzten Algorithmus erklären, seine asymptotische Laufzeit und seinen asymptotischen Speicherverbrauch analysieren und erklären, warum der Ansatz korrekt ist. Die Beschreibung des Algorithmus sollte auch verständlich sein, ohne seinen Quellcode zu lesen.
- **Programm.** Dieser Teil sollte den Quellcode der wichtigsten Teile des Algorithmus in Pascal, C oder C++ enthalten. Du kannst einfache Teile überspringen, wie zum Beispiel Input, Output, einfache mathematische Ausdrücke usw.

Beachte: Wenn ein Teil deiner Lösung für mehrere (Teil)aufgaben gilt, reicht es, ihn einmal aufzuschreiben und von da an auf ihn zu verweisen.





---

## Runde 2T

# Aufgabenbeschreibungen

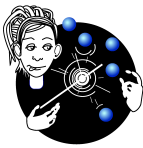
### 2T–1: Gerüchte

Maus Mark betreibt ein soziales Netzwerk namens Mousebook. Als Betreiber kennt er die komplette Netzwerkstruktur: Das bedeutet, er weiss, wer mit wem befreundet ist (und mit wem nicht). Informationen breiten sich im Netzwerk auf die folgende Art und Weise aus: Jede Maus, die eine neue Nachricht erhält, liest sie sich zuerst sorgfältig durch, und leitet sie dann sofort an alle ihre Freunde weiter. Nachrichten auf Mousebook kommen in dem Moment beim Empfänger an, in dem sie gesendet werden. Jede Maus auf Mousebook hat eine bestimmte Lesegeschwindigkeit, angegeben in (ganzen) Sekunden pro Buchstabe (Leerzeichen werden nicht gezählt). Aus vorangegangenen Interaktionen auf Mousebook hat Maus Mark diese Geschwindigkeit für jede Maus exakt bestimmen können.

Angestossen wird die Kommunikation dadurch, dass Maus  $X$  ein neues Gerücht erfindet und es allen ihren Freunden sendet.

Maus Mark interessiert sich nun dafür, wie schnell sich Informationen in seinem Netzwerk ausbreiten können. Das bedeutet, dass er voraussagen will, wie lange es nach dem initialen Versand dauern wird, bis alle Mäuse auf Mousebook das neue Gerücht gelesen haben. Wir nehmen hier an, dass dies immer in endlicher Zeit geschehen kann, und dass während der ganzen Zeit nur ein einziges Gerücht im Umlauf ist. Ausserdem liest und verschickt jede Maus ein Gerücht nur einmal; falls sie es ein weiteres Mal bekommt, ignoriert sie es einfach.

**Aufgabe** Bei Mousebook sind  $n$  Mäuse angemeldet. Für jede Maus sind ihre Lesegeschwindigkeit sowie eine Liste ihrer Freunde gegeben. Freundschaften sind immer gegenseitig. Ausserdem ist Maus  $X$  gegeben, sowie das Gerücht, welches sie verfasst hat. Nach wievielen Sekunden haben alle Mäuse das Gerücht gelesen?



Für die ersten zwei Teilaufgaben gibt es ein konkretes Beispiel, für welches du die Lösung bestimmen sollst; für jede Teilaufgabe sollst du ausserdem einen Algorithmus finden und beschreiben, der beliebige Beispiele lösen kann.

Beachte, dass du wie bei allen anderen Aufgaben von 2T die Implementierung von Input/Output weitgehend weglassen kannst. Darunter fallen beispielsweise die Berechnung der Länge des Gerüchts oder die Umwandlung der Freundschaftslisten in ein handlicheres Format. Du musst aber präzise angeben, welche Variable welchen Inputwert enthält.

**a) (30 Punkte)** Für diese Teilaufgabe nehmen wir an, dass alle Mäuse dieselbe Lesegeschwindigkeit haben, und diese 1 Sekunde pro Buchstabe beträgt.

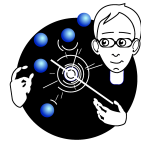
**Konkretes Beispiel (10 Punkte):**

Maus	s/B.	Freunde
Anna	1	Brigitte, Joachim, Stoff
Brigitte	1	Anna, Horst, Stoff
Charlie	1	Miriam, Stoff
Horst	1	Brigitte, Jessica, Josef
Jessica	1	Horst
Joachim	1	Anna, Josef, Richard
Josef	1	Horst, Joachim, Sophia
Miriam	1	Charlie, Sophia
Monika	1	Richard, Stoff
Richard	1	Joachim, Monika
Sophia	1	Josef, Miriam
Stoff	1	Anna, Brigitte, Charlie, Monika

Maus X: Monika

Gerücht: “Der Mond ist aus Schweizer Käse” (26 Buchstaben)

Für die vollständige Beschreibung des Algorithmus für diese Teilaufgabe (inklusive Erklärung/Korrektheitsbeweis/Kostenanalyse) erhältst du 20 Punkte.



**b) (50 Punkte)** Für diese Teilaufgabe gibt es keine weiteren Beschränkungen an die Lesegeschwindigkeiten.

**Konkretes Beispiel (10 Punkte):**

(Das Netzwerk ist dasselbe wie das des letzten Beispiels. Die Lesegeschwindigkeiten sind verschieden.)

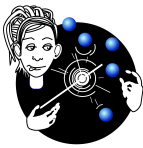
Maus	s/B.	Freunde
Anna	1	Brigitte, Joachim, Stoff
Brigitte	9	Anna, Horst, Stoff
Charlie	2	Miriam, Stoff
Horst	1	Brigitte, Jessica, Josef
Jessica	3	Horst
Joachim	7	Anna, Josef, Richard
Josef	2	Horst, Joachim, Sophia
Miriam	1	Charlie, Sophia
Monika	3	Richard, Stoff
Richard	5	Joachim, Monika
Sophia	3	Josef, Miriam
Stoff	1	Anna, Brigitte, Charlie, Monika

Maus  $X$ : Jessica

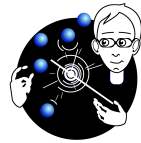
Gerücht: “Käse ist so gesund” (15 Buchstaben)

Für die vollständige Beschreibung aller Komponenten des Algorithmus für diese Teilaufgabe (inklusive Erklärung/Korrektheitsbeweis/Kostenanalyse) erhältst du 40 Punkte.

**c) (20 Punkte)** Kürzlich hat sich eine neue Maus  $Y$  im Netzwerk angemeldet, welche ebenso kreativ neue Gerüchte erfindet wie Maus  $X$ . Es kommt nun häufig vor, dass zur selben Zeit zwei Gerüchte unterwegs sind, eines, welches von Maus  $X$  verfasst wurde, sowie eines, welches von Maus  $Y$  verfasst wurde. Deshalb kann es jetzt auch passieren, dass eine Maus noch mit der Lektüre eines Gerüchts beschäftigt ist, wenn bereits ein neues Gerücht ankommt. In diesem Fall beginnt die Maus erst dann, das zweite Gerücht zu lesen, wenn sie das erste bereits weitergesendet hat. Falls zwei Gerüchte genau zur selben Zeit ankommen, dann wird zuerst das Gerücht von Maus  $Y$  gelesen, da Maus  $X$



einen langjährigen Ruf für gelegentliche irrelevante Falschmeldungen besitzt. Wir nehmen hier an, dass Maus  $X$  und Maus  $Y$  ihre Gerüchte gleichzeitig versenden und dass während der ganzen Zeit nur diese beiden Gerüchte im Umlauf sind. Erkläre, welche Modifikationen an Algorithmus/Korrektheitsbeweis/Kostenanalyse zu machen sind, um bestimmen zu können, wie lange es dauert, bis beide Gerüchte von allen Mäusen gelesen wurden.



## 2T–2: Abspülen

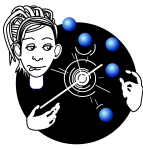
Maus Stoff arbeitet als Tellerwäscher in einem noblen Restaurant. Abends bekommt er einen Stapel dreckiger Teller, die er im Spülbecken waschen und dann in den Schrank stellen muss. Dabei muss er auf eine Besonderheit des Restaurants Rücksicht nehmen: Alle Teller sind unterschiedlich gross und er muss sie unbedingt als einen nach Durchmesser sortierten Stapel (der Grösste unten, der Kleinste oben) in den Schrank legen, da sonst die Kellner den Überblick verlieren würden. Ausserdem ist im Spülbecken nicht genug Platz; die Teller können dort nicht nebeneinander liegen und müssen auch dort einen Stapel bilden. Wir können also annehmen, dass Stoff ausschliesslich die folgenden zwei Schritte ausführen kann:

1. Er nimmt den Obersten aus dem Stapel dreckiger Teller und legt ihn oben auf den Stapel im Spülbecken. Da dieses mit einer kräftigen Spüllösung gefüllt ist, wird der Teller ohne weiteres Zutun von Stoff sofort sauber.
2. Er nimmt den zuletzt ins Spülbecken gelegten Teller und legt ihn oben auf den Stapel im Schrank.

(Andere denkbare Schritte, wie das Vertauschen von Tellern innerhalb des Stapels oder das Hochheben von mehreren Tellern auf einmal sind nicht möglich, da Stoff beide Arme braucht, um einen Teller zu stemmen – als Maus verfügt er nicht über viel Muskelkraft.)

Stoff kombiniert diese Schritte in beliebiger Reihenfolge um seine Aufgabe zu erfüllen (da er nicht unsterblich ist, müssen es aber endlich viele sein). Doch obwohl er auf eine beträchtliche Arbeitserfahrung zurückgreifen kann, gelingt ihm dies manchmal nicht; er vermutet sogar, dass es gar nicht möglich ist, egal wie klug er sich anstellt, wenn die dreckigen Teller in einer ungünstigen Abfolge auf dem Stapel liegen. Kannst du Stoff helfen, indem du ein Programm schreibst, dass ihm gleich sagt, wann er es mit einem solchen Stapel zu tun hat?

**Aufgabe** Sei  $n$  die Anzahl Teller auf dem “dreckigen” Stapel. Dein Programm soll als Eingabe eine Liste von  $n$  Zahlen entgegennehmen, welche die Durchmesser der Teller angibt. Dann soll es ausgeben, ob dieser Stapel mit



einer endlichen Abfolge der beiden oben beschriebenen Schritte sortiert in den Schrank gelegt werden kann.

**Eingabe** Die erste Zeile der Eingabe enthält die Anzahl  $n$  der dreckigen Teller. Die nächste Zeile enthält  $n$  Zahlen, die Durchmesser der Teller in ml (Mauslängen). Dabei gibt für  $1 \leq i \leq n$  die  $i$ -te Zahl den Durchmesser des  $i$ -ten Tellers von oben an. Du kannst davon ausgehen, dass die Durchmesser paarweise unterschiedlich sind.

**Ausgabe** Falls Stoff die Teller sortiert in den Schrank bekommen kann, soll dein Programm “YES” ausgeben, falls nicht, “NO”. Beachte, dass es sehr wichtig ist, zu begründen, wieso dein Programm korrekt ist.

BEISPIEL:

**Eingabe:**

7  
2.8 4.1 3.9 5.2 7.6 1.1 6.9

**Eingabe:**

8  
0.01 0.15 200 1 2 70.2 3.5 4

**Ausgabe:**

NO

**Ausgabe:**

YES





## 2T–3: Käseversteck

Maus Stoff lebt mit vielen anderen Mäusen in einem grossen Bau. Dieser Bau hat auch ein Versteck für den Käsevorrat ihres besten Käses. Dieses Versteck besteht aus einem Eingang, vielen Verzweigungen mit jeweils zwei Wegen und dem Käsevorrat am Ende eines der Wege. Damit die Mäuse den Weg zum Käse wieder finden, haben sie an jeder Verzweigung ein Wegweiser aufgehängt, das den Weg zum Vorrat angeben soll. Leider wurde das Versteck vor Kurzem renoviert und die Bauarbeiter haben alle Wegweiser nach links ausgerichtet. Da dies der Baufirma peinlich war, hat sie den Mäusen nichts davon gesagt; diese glauben immer noch, dass die Wegweiser zum Vorrat führen.

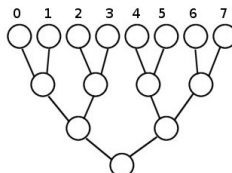
Jede Nacht hat eine Maus Nachtwache, macht sich aber mitten in der Nacht auf, um selber vom Käse zu naschen. Jede Maus folgt den Wegweisern, dreht jedoch alle Wegweiser auf ihrem Weg um, damit sie den Käse nur für sich hat.

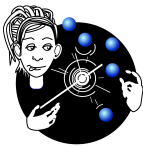
Finde heraus, ob Stoff den Käse findet.

**Aufgabe** Das Versteck besteht aus einem vollständigen binären Baum der Höhe  $h$ . Dies bedeutet, dass Stoff bei einer Verzweigung beginnt (dies ist der Eingang) und bei jeder Verzweigung nur zwischen einem linken und einem rechten Weg entscheiden kann. Ausserdem erreicht er das Ende des Weges immer nach genau  $h - 1$  Abzweigungen. Wir nummerieren die Blätter des Baumes (d.h. die Wegenden) mit 0 bis  $2^{h-1} - 1$ , wobei 0 das Blatt ist, welches erreicht wird, wenn man immer links geht.

Stoff hat in der  $n$ -ten Nacht Wache. Beachte, dass Stoff ein C-Programmierer ist und deshalb die Nächte auch von 0 an zählt.

Abbildung 1: Vollständiger binärer Baum mit  $h = 4$





**a) 10 Punkte** Sei für diese Teilaufgabe  $h = 4$  und  $n = 5$ . Wo muss der Käse liegen, damit Stoff ihn bekommt?

**b) 80 Punkte** Sei für diese Teilaufgabe  $n < 2^{h-1}$ . Schreibe einen Algorithmus, der bestimmt, bei welchem Blatt Stoff ankommt. Wie immer ist es wichtig, dass die Beschreibung vollständig ist (inkl. Laufzeitanalyse/Korrektheitsbegründung).

**c) 10 Punkte** Sei  $n$  beliebig (d.h. es kann auch grösser oder gleich  $2^{h-1}$  sein). Welche Modifikationen sind zu machen?



## 2T–4: Internat

Maus Rowan ist Rektor eines altehrwürdigen englischen Internats. Seit einiger Zeit plagen ihn schwere Regelverstösse in der Schulbibliothek: Gewisse Schüler nehmen sich Bücher aus der Bibliothek, ohne die Bibliothekskarte wie vorgeschrieben vorzuzeigen. Zwar wurde bisher jedes Buch wieder zurückgelegt, aber oftmals in falscher Reihenfolge.

Glücklicherweise wurde ein Missetäter auf frischer Tat ertappt. Um ihn und andere vor weiteren Zuwiderhandlungen gegen die Schulordnung abzuhalten, will Rowan ein Exempel statuieren. Die Strafe soll nach der Anzahl unrechtmässig mitgenommener Bücher bemessen werden, wobei sich Rowan gnädig zeigen möchte: Falls sich diese Anzahl nicht mit Sicherheit bestimmen lässt, will er die kleinstmögliche annehmen. Das heisst, dass er beweisen können will, dass mindestens soviele Bücher weggenommen (und dann wieder zurückgelegt) wurden. Dafür zählt er auf deine Hilfe!

**Aufgabe** In der Bibliothek stehen  $n$  Bücher. Anfangs sind sie sortiert und wir weisen ihnen der Einfachheit halber Zahlen von 1 bis  $n$  zu. Der Missetäter nimmt jeweils ein Buch und stellt es an einer anderen Stelle wieder ab. So wird beispielsweise aus der ursprünglichen Folge

$$12345 \quad (n = 5)$$

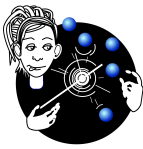
in einem Schritt (Buch 4 herausnehmen, vor Buch 1 zurücklegen) die Folge

$$41235$$

und in einem weiteren Schritt (Buch 3 herausnehmen, zwischen Buch 1 und Buch 2 zurücklegen)

$$41325.$$

Die Aufgabe kann also wie folgt beschrieben werden: Es sind  $n$  und eine Folge (also Permutation, falls dir dieser Begriff schon geläufig ist) von  $n$  Büchern gegeben. Wie viele Schritte sind mindestens nötig, um aus der ursprünglichen Folge (also der identischen Permutation) die gegebene zu erzeugen? Jede der folgenden Teilaufgaben enthält ein konkretes Beispiel, das du ausrechnen solltest; ausserdem solltest du einen Algorithmus finden, der beliebige Beispiele (unter den Annahmen der Teilaufgabe) lösen kann.



Beachte, dass in den ersten beiden Teilaufgaben zusätzliche Annahmen darüber getroffen werden, wohin ein Buch zurückgelegt werden darf. So ist beispielsweise der zweite Schritt des obigen Beispiels unter diesen Annahmen nicht zulässig.

**Eingabe** Die erste Zeile der Eingabe enthält eine natürliche Zahl  $n \geq 1$ , die Anzahl Bücher. Die nächste Zeile enthält die natürlichen Zahlen von 1 bis  $n$ , jeweils getrennt durch ein Leerzeichen, in beliebiger Reihenfolge; dies ist die Reihenfolge der Bücher.

**Ausgabe** Die Ausgabe soll genau eine Zahl enthalten, nämlich die Mindestanzahl Schritte, die nötig sind, um aus der ursprünglichen Reihenfolge der Bücher diejenige aus der Eingabe zu erzeugen.

**a) (30 Punkte)** Für diese Teilaufgabe nehmen wir an, dass ein weggenommenes Buch nur an den Anfang zurückgelegt werden kann. Im obigen Beispiel wäre also der erste Schritt zulässig, der zweite aber nicht.

**Konkretes Beispiel (5 Punkte):**

7  
6231457

Für die vollständige Beschreibung des Algorithmus für diese Teilaufgabe (inklusive Erklärung/Korrektheitsbeweis/Kostensanalyse) erhältst du 25 Punkte.

**b) (35 Punkte)** Für diese Teilaufgabe nehmen wir an, dass ein weggenommenes Buch entweder an den Anfang oder an das Ende zurückgelegt werden kann. Wiederum wäre der erste Schritt des obigen Beispiels zulässig, der zweite aber nicht. Allerdings wäre zum Beispiel  $41235 \rightarrow 41352$  als zweiter Schritt erlaubt (Buch 2 nehmen und an das Ende zurücklegen).

**Konkretes Beispiel (5 Punkte):**

7  
2534716



---

Für die vollständige Beschreibung des Algorithmus für diese Teilaufgabe (inklusive Erklärung/Korrektheitsbeweis/Kostensanalyse) erhältst du 30 Punkte.

**c) (35 Punkte)** In dieser Teilaufgabe gibt es keine Beschränkungen der erlaubten Schritte. Im obigen Beispiel wären also beide Schritte zulässig.

**Konkretes Beispiel (5 Punkte):**

7

3126574

Für die vollständige Beschreibung des Algorithmus für diese Teilaufgabe (inklusive Erklärung/Korrektheitsbeweis/Kostensanalyse) erhältst du 30 Punkte.