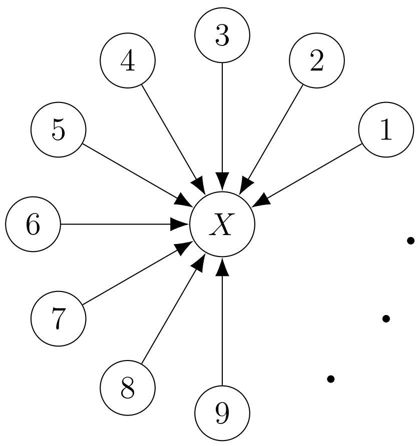
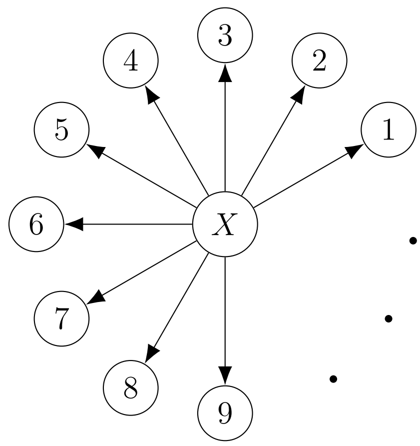
Teil (i). Jeder Roboter muss mindestens eine Nachricht senden, sonst kennt kein anderer Roboter dessen Informationen. Nach 999 gesendeten Nachrichten gibt es einen Roboter X, der noch gar keine Nachricht gesendet hat. Die Informationen von X müssen noch an alle 999 Roboter außer X selbst verteilt werden, wofür es pro Roboter mindestens eine Nachricht, insgesamt also 999 Nachrichten braucht. In der Summe ergeben sich mindestens $999 + 999 = 1\,998$ zu sendende Nachrichten.

Teil (ii). Da jeder Roboter eine ID hat und auf allen Robotern dasselbe Programm installiert ist, können wir schon bei der Programmierung eine beliebige ID wählen, den zugehörigen Roboter X zum *Koordinator* erheben und der Koordinator selbst wie auch die übrigen Roboter werden wissen, wer der Koordinator ist.

Im ersten Schritt senden alle Roboter (außer X selbst) eine Nachricht an X, das macht 999 Nachrichten (vgl. Abbildung 1a). X sendet anschließend eine Nachricht an jeden Roboter mit seinen eigenen Informationen und allen Informationen, die er bekommen hat (vgl. Abbildung 1b). Das erfordert 999 Nachrichten. Anschließend hat jeder Roboter alle Informationen. Insgesamt wurden 1 998 Nachrichten gesendet, was der unteren Schranke aus Teil (i) entspricht.



(a) Erste Phase



(b) Zweite Phase

Abbildung 1: Koordinatorbasierter Algorithmus zu Teil (ii)

Teil (iii). Der koordinatorbasierte Algorithmus aus Teil (ii) genügt hier nicht, denn in der zweiten Phase muss der Koordinator 999 Nachrichten senden und er benötigt dafür mindestens $999/3 = 333$ Tage. Ein anderer Algorithmus ist gesucht.

Erste Lösung. Die Roboter kommunizieren in einem bidirektionalen Pfad, das ist ein bidirektionaler Ring, der an einer Stelle „aufgebrochen" ist (vgl. Abbildung 2). Die IDs seien ohne Einschränkung die natürlichen Zahlen von 1 bis 1 000, notfalls integriert man das durch eine lange Fallunterscheidung fest in das Programm.

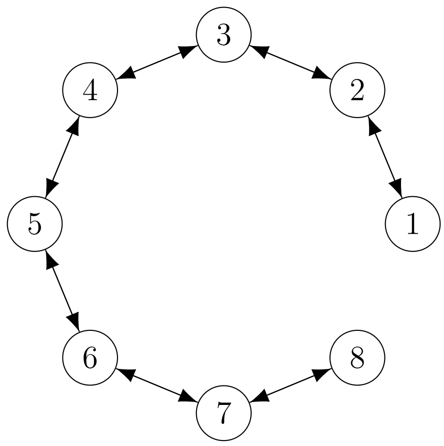


Abbildung 2: Kommunikation im bidirektionalen Pfad

Im Folgenden sprechen wir zur Abkürzung von „Roboter i" (für $1 \leq i \leq 1\,000$) und meinen damit den Roboter mit der ID i. Aus Gründen der Übersichtlichkeit zeigt Abbildung 2 die Situation für acht statt 1 000 Roboter.

Der Algorithmus arbeitet in zwei Phasen. In der ersten Phase sendet Roboter 1 eine Nachricht an Roboter 2, der wiederum eine Nachricht an Roboter 3 sendet usw. Allgemein ausgedrückt wartet Roboter i auf die Nachricht M von Roboter $i-1$ und sendet eine Nachricht M' mit allen Informationen aus M sowie seinen eigenen Informationen an Roboter $i+1$ (für $2 \leq i \leq 999$). Nach 999 gesendeten Nachrichten verfügt Roboter 1 000 über alle Informationen und die erste Phase ist abgeschlossen.

In der zweiten Phase läuft das gleiche Vorgehen ab, nur in umgekehrter Richtung: Roboter 1 000 sendet eine Nachricht an Roboter 999, der wiederum eine Nachricht an Roboter 998 sendet usw. Für $2 \leq i \leq 999$ wartet Roboter i auf die Nachricht von Roboter $i+1$, ergänzt seine eigenen Informationen und sendet eine Nachricht an Roboter $i-1$ weiter. Nach 999 Nachrichten ist die zweite Phase abgeschlossen.

Sei i ein beliebiger Roboter. In der ersten Phase erhält Roboter i eine Nachricht mit allen Informationen der Roboter $1, 2, \ldots, i-2, i-1$ und in der zweiten Phase eine Nachricht mit allen Informationen der Roboter $i+1, i+2, \ldots, 999, 1\,000$ sodass er am Ende über alle Informationen verfügt. Da in beiden Phasen je 998 Nachrichten gesendet werden, und jeder Roboter (außer 1 und 1 000 mit jeweils einer Nachricht) genau zwei Nachrichten sendet, erfüllt der Algorithmus alle Voraussetzungen führt zu einer korrekten Lösung. Tatsächlich kann er an einem einzigen Tag vollständig ausgeführt werden und benötigt damit deutlich weniger Zeit als zwei Wochen.

Zweite Lösung. Der Kommunikation nutzt die Struktur eines Ternärbaums aus, wie in Abbildung 3 dargestellt. Jeder Roboter befindet sich in einer bestimmten Schicht, z. B. ist Roboter 1 in Schicht 0, Roboter 3 in Schicht 1 und Roboter 7 in Schicht 2. Allgemein enthält Schicht k maximal 3^k Roboter. Im Falle von 1 000 Robotern

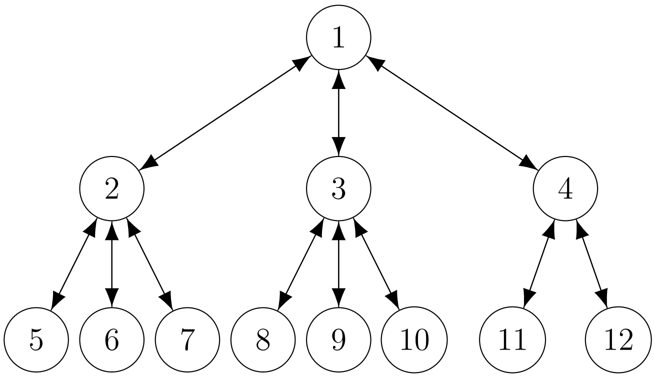


Abbildung 3: Kommunikation im Ternärbaum

gibt es genau 7 Schichten, denn $3^0 + 3^1 + 3^2 + 3^3 + 3^4 + 3^5 + 3^6 = 1\,093 \geq 1\,000$ und $3^0 + 3^1 + 3^2 + 3^3 + 3^4 + 3^5 = 364 < 1\,000$. Den Algorithmus beschreiben wir folgendermaßen. Im ersten Schritt senden alle Roboter aus Schicht 7 eine Nachricht an ihre zugehörigen Roboter aus Schicht 6. Dann senden die Roboter aus Schicht 6 Nachrichten an jene aus Schicht 5 usw. Am Ende verfügt Roboter 1 (aus Schicht 0) über alle Informationen und jeder Roboter außer Roboter 1 hat genau eine Nachricht gesendet, was insgesamt 999 Nachrichten ergibt. Im zweiten Schritt sendet Roboter 1 eine Nachricht mit allen Informationen an die Roboter 2, 3 und 4 aus Schicht 1. Diese senden die Nachricht dann weiter an alle Roboter aus Schicht 2 usw. Schließlich kommen die Nachrichten bei Schicht 7 an und alle Roboter sind mit Informationen versorgt. In dieser Phase sendet wegen der Ternärbaumstruktur kein Roboter mehr als 3 Nachrichten und da jeder Roboter außer Roboter 1 genau eine Nachricht empfängt, müssen auch 999 Nachrichten gesendet werden.

In der Summe werden 1 998 Nachrichten gesendet und der Algorithmus ist in zwei Tagen ausführbar, denn kein Roboter sendet mehr als 4 Nachrichten.

Bemerkung. Allgemein sind für n Roboter $2n - 2$ Nachrichten erforderlich und mit den obigen Lösungen auch hinreichend. Ändert man das Modell von Nachrichten auf z. B. Telefonanrufe, bei denen beide Kommunikationspartner vom jeweils anderen Partner Informationen empfangen, dann ist die untere Schranke $2n - 4$ Nachrichten.

Alternative Lösung oder Fehler gefunden?
Wir freuen uns über Verbesserungen und neue Lösungsideen.
Kontakt: **itag-goethe@protonmail.com**