26. Bundeswettbewerb Informatik 2007/2008
Endrunde 23.-26.9.2008

Räuber und Polizisten

Gruppenarbeit 2. Tag

In der Informatik spielen *Graphen* eine große Rolle; ein Graph G ist eine mathematische Struktur, bestehend aus einer Menge von *Knoten* V und einer Menge von *Kanten* E. Eine Kante ist ein Knotenpaar, so dass man sich eine Kante als Verbindung zweier Knoten vorstellen kann. Ein *Weg* zwischen zwei Knoten setzt sich aus einer oder mehreren Kanten zusammen. Die Abbildungen 1 bis 4 zeigen verschiedene Graphen.

Räuber und Polizisten ist ein Spiel, das von 2 Parteien (dem *Räuber* und den *Polizisten*) auf einem zusammenhängenden Graphen gespielt wird: Gegeben ist ein ungerichteter zusammenhängender Graph (d.h. die Kanten haben keine Pfeilrichtung, und jeder Knoten ist von jedem anderen Knoten aus durch einen Weg erreichbar) $G = (V, E)$ und eine natürliche Zahl $k \geq 1$. Im Räuber-und-Polizisten-Spiel für G und k soll ein Räuber von k Polizisten gefangen werden, die alle auf den Knoten von G leben.

Für dieses Spiel gibt es folgende Regeln:

Bewegung der Figuren (a) Jeder der *Polizisten* kann in einem Hubschrauber von Knoten zu Knoten fliegen. Es gibt eine Leitstelle, die sieht, wo der Räuber gerade ist, und die den Polizisten sagt, wo sie als nächstes hinfliegen sollen. (b) Der *Räuber* hat keinen Hubschrauber zur Verfügung, sondern muss sich entlang der Kanten von G bewegen. Er weiß, wo die Polizisten gerade sind, und da er den Polizeifunk abhört, weiß er auch, wo sie als nächstes hinfliegen wollen, und versucht, während ihres Fluges zu entkommen. Er kann beliebig schnell und beliebig weit laufen, darf dabei aber keinen der Knoten betreten, die vor *und* nach dem Flug von einem Polizisten besetzt sind.

Beginn des Spiels Zu Beginn des Spiels wählt die Leitstelle eine Menge von Knoten aus, die von den k Polizisten besetzt werden. Danach wählt der Räuber einen nicht von Polizisten besetzten Knoten als seine Anfangsposition aus.

Spielrunden Das Spiel wird in Runden gespielt, wobei jede Runde wie folgt verläuft: Sei P die Menge der Knoten, die zu Beginn der Runde von den Polizisten besetzt sind, und sei r der Knoten, an dem sich der Räuber befindet; (P, r) ist also die *Spielposition* zu Beginn einer Runde. Die Polizisten steigen nun in ihre Hubschrauber und fliegen zu neuen Knoten P'. Dabei ist es durchaus erlaubt, dass einige Polizisten einfach da bleiben, wo sie gerade sind, d.h., es ist erlaubt, dass $P \cap P' \neq \emptyset$. Noch während des Flugs der Polizisten flüchtet der Räuber zu einem Knoten r', der von seinem derzeitigen Standort r aus durch einen Weg in G erreichbar ist. Dieser Weg darf aber nicht durch einen Knoten führen, der in $P \cap P'$ liegt. Es ist allerdings erlaubt, dass der Räuber einfach da stehen bleibt, wo er ist, d.h. er darf $r' := r$ wählen. Die Runde wird dadurch beendet, dass die Hubschrauber landen. Die

Polizisten befinden sich nun auf den Knoten in P', und der Räuber befindet sich auf Knoten r'. Falls $r' \in P'$ ist, haben die Polizisten den Räuber gefangen und das Spiel ist beendet. Falls $r' \notin P'$ ist, wird eine weitere Runde gespielt.

Ziel des Spiels Ziel der Polizisten ist natürlich, den Räuber nach endlich vielen Runden zu fangen, während es Ziel des Räubers ist, immer wieder zu entkommen.

Sind ein Graph G und eine Zahl k gegeben, wollen wir herausfinden, ob – und falls ja, wie – die Polizisten den Räuber fangen können.

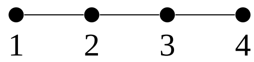

Abbildung 1: Graph mit vier Knoten

Beispiel 1 Ist G der Graph aus Abbildung 1, so kann der Räuber von 2 Polizisten gefangen werden: Zunächst besetzen die Polizisten die Knoten 1 und 2. Der Räuber muss daher als Anfangsposition einen der Knoten 3 oder 4 wählen. In der ersten Runde fliegen die Polizisten zu den Knoten 2 und 3. Da der Knoten 2 vor *und* nach dem Flug besetzt ist, darf der Räuber auf seinem Fluchtweg den Knoten 2 nicht betreten. Daher befindet er sich am Ende der ersten Runde entweder auf Knoten 3 und ist somit gefangen, oder er befindet sich auf Knoten 4 und es geht mit der zweiten Runde weiter. In der zweiten Runde fliegen die Polizisten zu den Knoten 3 und 4. Da der Knoten 3 vor *und* nach dem Flug besetzt ist, darf der Räuber auf seinem Fluchtweg den Knoten 3 nicht betreten. D.h. er kann sich nicht vom Knoten 4 wegbewegen und ist daher am Ende der Runde gefangen.

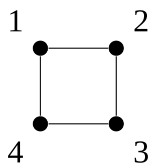

Abbildung 2: Graph mit vier Knoten

Beispiel 2 Auf dem in Abbildung 2 gegebenen Graphen G können 3 Polizisten den Räuber ganz leicht fangen, indem sie folgendermaßen vorgehen: Am Anfang besetzen die Polizisten die Knoten 2, 3 und 4. Als Startposition muss der Räuber daher den Knoten 1 wählen. In der ersten Runde fliegen die Polizisten zu den Knoten 1, 2 und 4. Da die Knoten 2 und 4 sowohl vor als auch nach dem Flug von Polizisten besetzt sind, kann der Räuber sich nicht vom Knoten 1 wegbewegen und ist daher am Ende der ersten Runde gefangen.

Stehen jedoch nur 2 Polizisten zur Verfügung, so kann der Räuber immer wieder entkommen: Als Startposition r wählt er einen beliebigen Knoten, der nicht zu den von den Polizisten besetzten Knoten P gehört (einen solchen Knoten gibt es, weil P höchstens 2 Knoten enthält, der Graph aber aus mehr als 2 Knoten besteht). In jeder einzelnen Runde bewegt sich der Räuber dann gemäß der folgenden Strategie (r, P und P' haben die gleiche Bedeutung wie in den Spielregeln): Wir betrachten zwei Fälle. *Fall 1:* $r \notin P'$. In diesem Fall bleibt der Räuber einfach da stehen, wo er ist, und es ist klar, dass er dann in dieser Runde nicht gefangen wird. *Fall 2:* $r \in P'$. In diesem Fall wissen wir, dass mindestens einer der beiden Nachbarn von Knoten r *nicht* zu P' gehört (da es nur 2 Polizisten gibt, besteht P' nämlich aus maximal 2 Knoten, und einer davon ist der Knoten r selbst). Zu diesem Nachbarn flüchtet sich der Räuber – und es ist klar, dass er in dieser Runde nicht gefangen wird.

Aufgabe 1 Für jede natürliche Zahl $n \geq 4$ sei G_n der Graph, der n Knoten zu einem Kreis verbindet. Beispielsweise zeigt Abbildung 1 den Graphen G_4.

Betrachten Sie zunächst den Graphen G_9 und geben Sie eine Strategie an, wie der Räuber in diesem Graphen von 3 Polizisten gefangen werden kann. Versuchen Sie dabei, die Zahl der Runden, die maximal gespielt werden müssen, bis der Räuber gefangen ist, so klein wie möglich zu halten.

Beschreiben Sie, wie für beliebiges $n \geq 4$ im Graphen G_n der Räuber von 3 Polizisten gefangen werden kann. Überlegen Sie auch, warum der Räuber im Graphen G_n stets entkommen kann, wenn nur 2 Polizisten zur Verfügung stehen.

Eine weitere Definition: Für einen beliebigen Graphen G sei $PW(G)$ die *kleinste* Zahl k von Polizisten, mit denen der Räuber im Graphen G gefangen werden kann. Die Zahl $PW(G)$ wird die *Polizistenweite* von G genannt. Als Ergebnis von Aufgabe 1 haben Sie beispielsweise gezeigt, dass für jedes $n \geq 4$ der Graph G_n die Polizistenweite $PW(G_n) = 3$ hat.

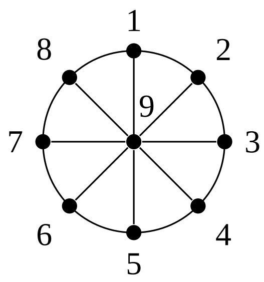

Abbildung 3: Regenschirm-Graph

Aufgabe 2 Was ist die Polizistenweite des Graphen aus Abbildung 3?

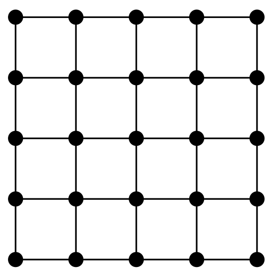

Abbildung 4: 5×5-Gitter-Graph

Aufgabe 3 Für jede Zahl $n \geq 2$ sei $G_{n\times n}$ der Graph, der aus dem $n \times n$-Gitter besteht. $G_{5\times 5}$ beispielsweise ist in Abbildung 4 skizziert. Bestimmen Sie die Polizistenweite des Graphen $G_{5\times 5}$ und überlegen Sie sich allgemein für $n \geq 2$, welche Polizistenweite der Graph $G_{n\times n}$ hat.

In der nächsten Aufgabe geht es darum, Fluchtstrategien für den Räuber zu finden. Dazu benötigen wir die folgende Definition: Sei $k \geq 1$. Ein Graph $G = (V, E)$ heißt *k-zusammenhängend*, wenn er mindestens $k+1$ Knoten besitzt und wenn für alle $V' \subseteq V$ mit $|V'| \leq k$ gilt: Der Graph, der aus G entsteht, indem sämtliche Knoten aus V' gelöscht werden (und sämtliche Kanten, die in einem Knoten aus V' enden), ist zusammenhängend.

Aufgabe 4 Sei $k \geq 1$ und sei G ein beliebiger k-zusammenhängender Graph. Beschreiben Sie eine Strategie, mit der der Räuber im Graphen G gegen k Polizisten stets entkommen kann.

Geben Sie ein oder mehrere Beispiele für k-zusammenhängende Graphen an. Können Sie auch ein Beispiel für einen Graphen angeben, der *nicht* k-zusammenhängend ist, bei dem der Räuber aber trotzdem k Polizisten stets entwischen kann?

Als nächstes sollen Verfolgungsstrategien für Polizisten automatisch generiert werden:

Aufgabe 5 Entwickeln Sie einen möglichst effizienten Algorithmus, der bei Eingabe eines Graphen G und einer Zahl k herausfindet, ob der Räuber auf dem Graphen G von k Polizisten stets gefangen werden kann. Analysieren Sie die Laufzeit Ihres Algorithmus (in Abhängigkeit von der Zahl k und der Größe n des Graphen).

Überlegen Sie sich auch einen Algorithmus, der eine "Verfolgungsstrategie" für die Polizisten berechnet. D.h.: Falls der Räuber von k Polizisten gefangen werden kann, soll der Algorithmus eine Tabelle berechnen, die zu jeder Spielposition (P, r) die Menge P' der Knoten angibt, zu denen die Polizisten in dieser Runde fliegen sollen. Die Tabelle soll natürlich garantieren, dass unabhängig von der Startposition und von den Bewegungen des Räubers der Räuber nach einer endlichen Anzahl von Runden gefangen ist.

Zum Schluss noch eine Bemerkung dazu, warum das Räuber-und-Polizisten-Spiel interessant für Informatiker/innen ist: Es sind viele Probleme bekannt, die auf allgemeinen Graphen "schwer" zu lösen (genauer gesagt: NP-schwierig) sind, die aber sehr effizient gelöst werden können, wenn man weiß, dass als Eingabe nur Graphen auftauchen, deren Polizistenweite $\leq k$ ist. Diese Algorithmen verfahren oft rekursiv entlang einer vorher ermittelten Verfolgungsstrategie, mit deren Hilfe k Polizisten den Räuber im gegebenen Graphen fangen können.

Falls Sie noch etwas Zeit übrig haben, können Sie sich noch überlegen, wie man die folgende Aufgabe lösen kann.

Aufgabe 6 Beim *3-Färbbarkeitsproblem* bekommt man als Eingabe einen Graphen G. Das Ziel ist, herauszufinden, ob die Knoten von G so mit den 3 Farben Rot, Gelb und Blau gefärbt werden können, dass Knoten, die durch eine Kante benachbart sind, unterschiedliche Farben haben. Es ist bekannt, dass dieses Problem NP-vollständig ist. Lässt man als Eingaben jedoch nur solche Graphen zu, die Polizistenweite $\leq k$ (für ein festes k) haben, kann das 3-Färbbarkeitsproblem in Polynomialzeit (genauer gesagt in Zeit $\leq c_k \cdot n^2$, wobei c_k nur von k und nicht von der Größe n des eingegebenen Graphen abhängt) gelöst werden. Überlegen Sie, wie ein solcher Algorithmus vorgehen könnte. Betrachten Sie zunächst die Spezialfälle $k = 2$, $k = 3$ und $k = 4$ und beschreiben Sie danach, wie ein Algorithmus für beliebiges $k \geq 2$ vorgehen könnte.