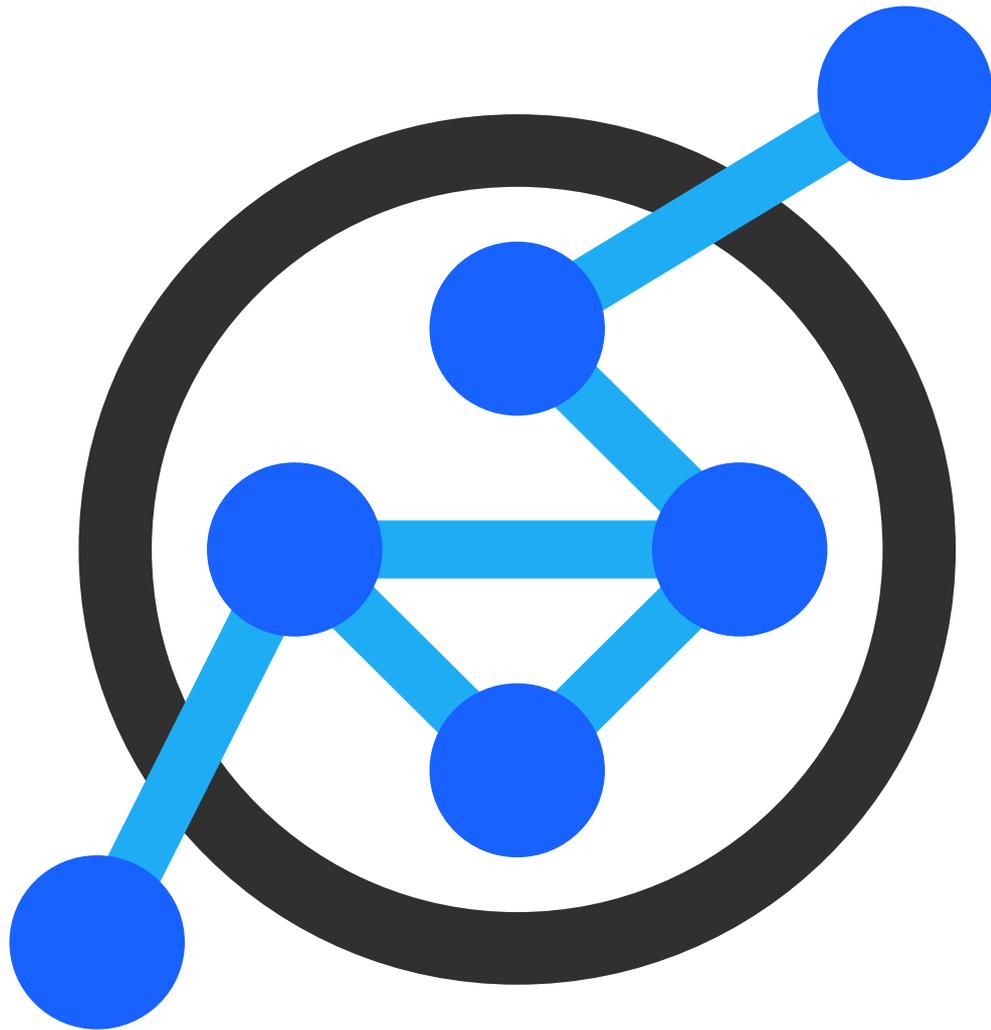


Fortgeschrittene Theorieaufgaben SOI Workshop



Swiss Olympiad in Informatics

Oktober 2020

Netzwerkverkabelung

Maus Stofl ist für die Netzwerkverkabelung an der SOI Finalrunde zuständig. Es stehen ihm hierfür Ethernetkabel unterschiedlicher Länge zur Verfügung. In der Mitte des Raumes befindet sich ein Switch. Rund um den Switch herum sind die Computer der Teilnehmer positioniert. Stofl muss jeden Computer über ein Ethernetkabel direkt mit dem Switch verbinden. Hierfür muss das Kabel natürlich mindestens so lange sein, wie die Distanz vom Computer zum Switch. Er fragt sich nun, ob dies mit den gegebenen Kabel möglich ist, und welches Kabel er für welchen Computer verwenden soll.

Formale Beschreibung Gegeben sind N Ethernetkabel der Länge l_0, \dots, l_{N-1} Meter. Maus Stofl muss M Computer mit einem Switch verbinden. Der i -te Computer ist d_i Meter vom Switch entfernt. Damit ein Computer über ein Kabel mit dem Switch verbunden werden kann, muss das Kabel mindestens so lange sein, wie die Distanz zum Switch.

Teilaufgabe 1: Algorithmus konzipieren

Erstelle einen Algorithmus für Stofl, der für gegebene Ethernetkabel und Distanzen zum Switch bestimmt, ob die Computer mit dem Switch verbunden werden können, und falls ja, welches Kabel für welchen Computer verwendet werden soll.

Argumentiere, warum dein Algorithmus *korrekt* ist und implementiere ihn in *Pseudocode*. Analysiere die *asymptotische Laufzeit* (wie viel Zeit in Abhängigkeit von N und M dein Algorithmus braucht) deines Algorithmus.



Käse-Kongress

Am Internationalen Käse-Kongress nehmen Mäuse aus M verschiedenen Regionen teil. Die Regionen sind von 0 bis $M - 1$ durchnummeriert. Aus der i -ten Region stammen s_i Experten für Käse. Die Teilnehmer des Kongresses werden nach Tradition in Zimmer der Grösse K untergebracht. Um den Austausch zu fördern, will das Organisationskomitee des Internationalen Käse-Kongresses die Teilnehmer nach einem bestimmten Schema auf die Zimmer aufteilen: In jedem Zimmer sollten Experten aus K unterschiedlichen Regionen untergebracht werden. Kannst du dem Organisationskomitee dabei helfen?

Formale Beschreibung Gegeben sind die Anzahl s_0, \dots, s_{M-1} Experten, die die einzelnen Regionen an den Kongress schicken, sowie die Grösse K der Zimmer. In jedem Zimmer sollten genau K Experten aus unterschiedlichen Regionen untergebracht werden. Es ist garantiert, dass die Summe S der s_i durch K teilbar ist (sodass die Experten genau auf $\frac{S}{K}$ Zimmer aufgeteilt werden).

Teilaufgabe 1: Algorithmus konzipieren

Erstelle einen Algorithmus für das Organisationskomitee, der entscheidet, ob es möglich ist, die Experten auf die Zimmer aufzuteilen. Fall dies möglich ist, sollte dein Algorithmus ausserdem für jedes Zimmer ausgeben, aus welchen Regionen die Mäuse in diesem Zimmer stammen.

Argumentiere, warum dein Algorithmus *korrekt* ist und implementiere ihn in *Pseudocode*. Begründe insbesondere, weshalb es nicht möglich ist, die Experten auf die Zimmer aufzuteilen bzw. weshalb bei deiner Aufteilung die Experten im gleichen Zimmer aus unterschiedlichen Regionen stammen. Analysiere die *asymptotische Laufzeit* (wie viel Zeit in Abhängigkeit von M und S dein Algorithmus braucht) deines Algorithmus.



Jurte

Die Nomadenvölker Kasachstans leben in Jurten. Eine Jurte ist ein traditionelles, grosses, rundes Zelt. Auf deiner Reise entdeckst du einen grossen Lagerplatz dessen N Plätze von 0 bis $N - 1$ durchnummeriert sind. Auf jedem Platz steht genau eine Jurte. Auch die Jurten sind nummeriert von 0 bis $N - 1$, sodass eigentlich immer ein Jurte und ein Zeltplatz zusammengehört. Doch weil die Jurten mitten in der Nacht in aller Eile aufgestellt wurden, sind die Jurten durcheinander gekommen und stehen nun nicht unbedingt auf dem Zeltplatz mit der gleichen Nummer. Deine Aufgabe ist es nun, die Jurten wieder in die richtige Reihenfolge zu bringen. Dazu kannst du Schritt für Schritt immer zwei beliebige Jurten abbauen, austauschen und wieder aufbauen. Du kannst nie mehr als zwei Jurten auf einmal abbauen, also immer nur zwei Jurten austauschen. Dein Ziel ist es mit so wenig Vertauschungen wie möglich alle Jurten wieder in die richtige Reihenfolge zu bringen.

Formale Beschreibung Gegeben sind N Ganzzahlen p_0, \dots, p_{N-1} , wobei p_i die Nummer der Jurte, die momentan auf Platz i steht, ist.

Teilaufgabe 1: Höchstens $N - 1$ Vertauschungen

Die Steppenmaus Stefl behauptet nun, dass es immer möglich ist N Jurten mit weniger als N (also höchstens $N - 1$) Vertauschungen in die richtige Reihenfolge zu bringen. Beschreibe einen Algorithmus, der die Jurten mit höchstens $N - 1$ in die richtige Reihenfolge bringt.

Argumentiere, warum dein Algorithmus *korrekt* ist und implementiere ihn in *Pseudocode*. Analysiere die *asymptotische Laufzeit* (wie viel Zeit in Abhängigkeit von N dein Algorithmus braucht) deines Algorithmus.

Teilaufgabe 2: Optimale Anzahl Vertauschungen

Du bist noch nicht zufrieden damit, dass es dir in weniger als N Vertauschungen gelingt, denn für manche Startaufstellungen geht es auch deutlich besser. Beschreibe einen Algorithmus, der die Jurten mit der minimalen Anzahl Vertauschungen in die richtige Reihenfolge bringt.

Argumentiere, warum dein Algorithmus *korrekt* ist und implementiere ihn in *Pseudocode*. Du sollst insbesondere begründen, dass jede erdenkliche Lösungsstrategie mindestens so viele Vertauschungen benötigt wie deine. Analysiere die *asymptotische Laufzeit* (wie viel Zeit in Abhängigkeit von N dein Algorithmus braucht) deines Algorithmus.



Fluss

In Kasachstan gibt es einige grosse Flüsse. So ist der Fluss Irtysch beispielsweise über 4000 Kilometer lang. Er entspringt in China, fliesst durch Kasachstan und mündet dann in Russland in den Ob. Entlang des Irtysch gibt es zahlreiche Städte. Sie sind von 0 bis $N - 1$ nummeriert. Jede Stadt hat zwei Stadtteile, einen Teil auf der linken und einen auf der rechten Flussseite, mit einer Brücke dazwischen, welche die beiden Stadtteile verbindet. Bei einer Überschwemmung sind nun leider alle N Brücken eingestürzt.

Um nun übergangsweise wieder Verbindungen über den Fluss herzustellen sind zwischen einigen gegebenen Paaren von Städten Fährverbindungen geplant. Damit die Fähren in den Städten anlegen können, muss jede Stadt genau eine Schiffsanlegestelle bauen. Eine solche Schiffsanlegestelle kann jeweils auf der linken oder rechten Flussseite der Stadt gebaut werden. Es soll jede Stadt eine solche Schiffsanlegestelle errichten, selbst wenn es gar keine Fährverbindung dahin gibt.

Neue Brücken über den Fluss sind vorerst keine geplant. Wenn man also den Fluss überqueren möchte, muss man zwingend die Fähre benützen. Damit es trotzdem möglichst einfach ist den Fluss zu traversieren, sollen möglichst viele Fährverbindungen über den Fluss fahren, das heisst auf zwei verschiedenen Flussseiten an- und ablegen. Eine Fährverbindung von Stadt A zu Stadt B überquert zum Beispiel den Fluss, wenn die Anlegestelle in A auf der linken und in B auf der rechten Flussseite gebaut wird. Wenn aber beide Anlegestellen auf der gleichen Seite gebaut werden, dann überquert die Fähre den Fluss nicht.

Nun müssen sich die Stadtpräsidentinnen und Stadtpräsidenten entscheiden auf welcher Flussseite sie ihre Anlegestelle bauen möchten. Kannst du ihnen helfen die Anlegestellen so zu bauen, dass mindestens die Hälfte der Fährverbindungen den Fluss überqueren?

Formale Beschreibung Gegeben sind N Städte und M Fährverbindungen. Die i -the Fährverbindung ist zwischen den Städten a_i und b_i geplant.

Teilaufgabe 1: Mindestens die Hälfte

Erstelle einen Algorithmus, der entscheidet, welche Anlegestellen auf der linken und welche Anlegestellen auf der rechten Seite gebaut werden sollten, sodass mindestens die Hälfte der Fährverbindungen den Fluss überqueren.

Argumentiere, warum dein Algorithmus *korrekt* ist und implementiere ihn in *Pseudocode*. Begründe insbesondere, weshalb bei deiner Ausgabe mindestens die Hälfte der Fährverbindungen den Fluss überqueren. Analysiere die *asymptotische Laufzeit* (wie viel Zeit in Abhängigkeit von N und M dein Algorithmus braucht) deines Algorithmus.