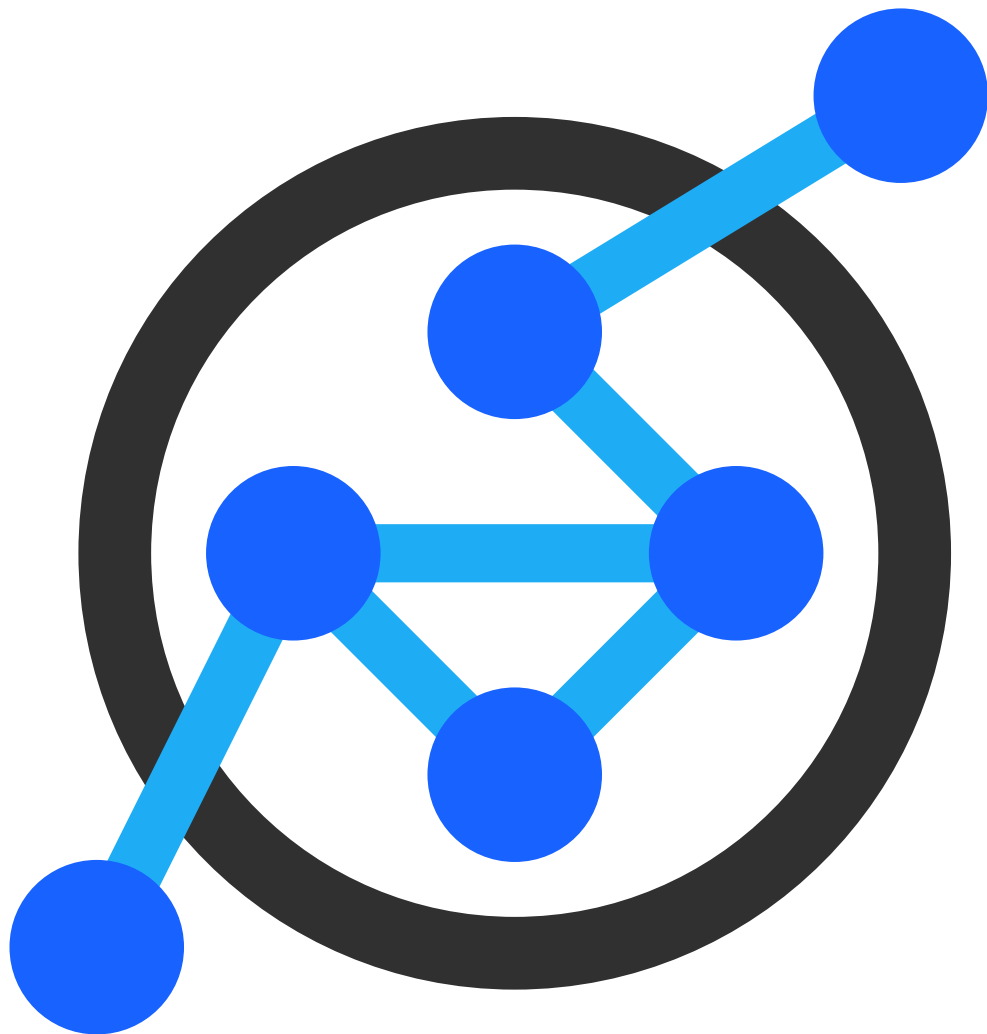


Zweite Runde Theorie

Aufgaben



Swiss Olympiad in Informatics

7. März 2020



Anweisungen

- Öffne die Prüfungs erst, wenn du dazu aufgefordert wirst. Die Prüfung beginnt für alle Teilnehmer gleichzeitig und dauert 5 Stunden.
- Mit Ausnahme von Uhren (keine Smartwatches) sind keine elektronischen Geräte auf den Tischen erlaubt. Schalte dein Mobiltelefon aus.
- Beginne jede Aufgabe auf einem neuen Blatt und schreibe deinen Namen auf alle Blätter. Nummeriere deine Seiten und sortiere sie vor der Abgabe.
- Verwende keinen Bleistift und schreibe nicht mit rot.
- Schreibe lesbar.

Bewertung

Deine Lösung wird anhand ihrer Korrektheit und der asymptotische Laufzeit und Speichernutzung bewertet, sowie der Begründung dieser Punkte. Wir erwarten, dass du einen Beweis oder eine Beweisskizze für die Korrektheit und Laufzeit/Speichernutzung lieferst.

Du kannst jederzeit auf den Inhalt des SOI Wikis und 2H verweisen. Du musst nicht erklären, wie z.B. Dijkstra funktioniert, aber solltest begründen, wo und wieso man ihn anwenden kann. Im Fall von Dijkstra muss der Graph klar definiert sein und seine Kantengewichte dürfen nicht negativ sein.

Der Algorithmus sollte genug detailliert beschrieben sein, dass man ihn anhand der Erklärung programmieren könnte. Wichtig ist hier auch, dass alle verwendeten Datenstrukturen genau beschrieben werden. Am einfachsten ist es erfahrungsgemäss, kurz einen Pseudocode hinzuschreiben.

Wir empfehlen dir, dich an folgende Struktur zu halten:

1. Beschreibe die Idee hinter deinem Algorithmus so verständlich wie möglich.
2. Gib Pseudocode oder erkläre, wie man den Algorithmus implementieren könnte.
3. Begründe, wieso der Ansatz die Aufgabe korrekt löst.
4. Analysiere die asymptotische Laufzeit und den asymptotischen Speicherverbrauch.

Wenn ein Teil deiner Lösung für mehrere Teilaufgaben gilt, reicht es, ihn nur einmal aufzuschreiben und von da an auf ihn zu verweisen. Beachte aber, dass ein allgemeiner Algorithmus die vorherigen Teilaufgaben zwar lösen kann, aber nicht zwingend optimal ist.

Viel Erfolg!

Verdunstung

Maus Binna freut sich, an der IOI 2020 in Singapur teilnehmen zu können. Als sie jedoch angekommen ist, realisiert sie, dass es in Singapur viel zu heiss ist für eine kleine Maus wie sie. Um nicht infolge von Dehydration zu sterben, kauft Maus Binna n Gläser mit Wasser. Das i -te Glas enthält a_i Milliliter Wasser. Infolge des heissen Wetters verdunstet jedoch jede Minute ein Milliliter Wasser aus jedem Glas. Da Maus Binna nicht die ganze Zeit zum Supermarkt gehen will, um Wasser zu kaufen, möchte sie das Wasser umgiessen. Zu Beginn jeder Minute giesst Maus Binna alles Wasser von einem Glas in ein anderes Glas. Hilf Maus Binna, die Menge an Wasser nach n Minuten zu maximieren.

Formale Beschreibung Gegeben ist eine Liste von n nichtnegativen Zahlen a_0, \dots, a_{n-1} . Jede Minute passiert das Folgende: Du kannst a_i auf $a_i + a_j$ und a_j auf 0 setzen (für i und j deiner Wahl), und alle Zahlen $a_i > 0$ werden zu $a_i - 1$. Finde die Maximale Summe aller Zahlen nach n Minuten.

Teilaufgabe 1: Ein Beispiel lösen (10 Punkte)

Gegeben ist die folgende Liste: $\{1, 2, 3, 4\}$. Was ist die maximale Menge an Wasser nach $n = 4$ Minuten? Gib alle Operationen und die Zeit, zu der du diese Aktionen machst, an, die benötigt werden, um diesen Wert zu erreichen. Du musst deine Antwort nicht begründen.

Teilaufgabe 2: Ganz Gleich (20 Punkte)

Alle Gläser beinhalten zu Beginn die selbe Menge an Wasser: genau k Milliliter ($a_i = k$ für alle $0 \leq i < n$). Gib einen Algorithmus an um die maximale Menge an Wasser für diesen Spezialfall zu berechnen (und beweise dessen Korrektheit).

Teilaufgabe 3: Permutationen (30 Punkte)

Alle Zahlen a_i formen eine Permutation der Zahlen $\{1, 2, \dots, n\}$. Gib einen Algorithmus für diesen Spezialfall an (und beweise dessen Korrektheit).

Teilaufgabe 4: Allgemeiner Fall (40 Punkte)

Für diese Teilaufgabe gibt es keine Einschränkungen für die Werte a_i . Gib einen Algorithmus für den allgemeinen Fall an und beweise dessen Korrektheit.

Zahlenjagd

Maus Stoffl macht an einer Zahlenjagd mit, bei der das Ziel ist, eine fehlende Zahl zu finden. Es gibt eine geheime Liste, die Stoffl nicht sehen kann, auf der $2^n - 1$ verschiedene Zahlen von 0 bis und mit $2^n - 1$ notiert sind. Folglich fehlt genau eine Zahl. Hilf Stoffl, die fehlende Zahl zu finden um die Zahlenjagd zu gewinnen.

Jede Zahl in der geheimen Liste kann als Folge von n Bits geschrieben werden. Zum Beispiel ist $6_{(10)} = 4+2 = 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 110_{(2)}$ oder $13_{(10)} = 8+4+1 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_{(2)}$. Das 0-te Bit ist der Koeffizient von 2^0 (d.h. 1 falls die Zahl ungerade und 0 falls sie gerade ist), das 1-te Bit ist der Koeffizient von 2^1 , und so weiter.

Es stehen Stoffl $(2^n - 1) \cdot n$ mögliche Quests zur Auswahl. Lösen von Quest $q(i, j)$ verrät ihm das j -te Bit der i -ten Zahl. Zum Beispiel in der Liste unten, Quest $q(5, 2)$ ergibt 1 und Quest $q(0, 2)$ ergibt 0. Quests werden nacheinander gelöst, d.h. Stoffl kann anhand von den Ergebnissen der vorhergehenden Quests entscheiden, welchen Quest er als nächstes lösen sollte.

Hilf Stoffl, die Zahlenjagd mit der minimalen Anzahl Quests zu lösen!

Num	Bits		
$a_0 = 3$	0	1	1
$a_1 = 1$	0	0	1
$a_2 = 6$	1	1	0
$a_3 = 7$	1	1	1
$a_4 = 0$	0	0	0
$a_5 = 4$	1	0	0
$a_6 = 2$	0	1	0

Beispiel einer geheimen Liste

Num	b_2	b_1	b_0
a_0	$q(0, 2)$	$q(0, 1)$	$q(0, 0)$
a_1	$q(1, 2)$	$q(1, 1)$	$q(1, 0)$
a_2	$q(2, 2)$	$q(2, 1)$	$q(2, 0)$
a_3	$q(3, 2)$	$q(3, 1)$	$q(3, 0)$
a_4	$q(4, 2)$	$q(4, 1)$	$q(4, 0)$
a_5	$q(5, 2)$	$q(5, 1)$	$q(5, 0)$
a_6	$q(6, 2)$	$q(6, 1)$	$q(6, 0)$

Liste möglicher Quests von Stoffl

Formale Beschreibung Sei $a_0, a_1, \dots, a_{2^n-1}$ eine Permutation der Zahlen $\{0, 1, \dots, 2^n - 1\}$. Du kannst Queries $q(i, j)$ stellen, für $0 \leq i \leq 2^n - 2$ und $0 \leq j < n$. Query $q(i, j)$ verrät dir das j -te Bit der i -ten Zahl (beachte, dass du nicht die $(2^n - 1)$ -te Zahl abfragen kannst). Die Queries werden nacheinander gestellt, d.h. du kann anhand von den Ergebnissen der vorhergehenden Queries entscheiden, welche Query du als nächste stellst. Finde den Wert a_{2^n-1} mit der minimalen Anzahl Queries.

Teilaufgabe 1: Analysiere Stoffls Liste (10 Punkte)

Stoffl hat bereits einige der Quests gelöst und folgende Liste erstellt:

Num	b_2	b_1	b_0
a_0		1	0
a_1	1		
a_2	1		1
a_3		1	0
a_4			0
a_5	1		1
a_6			

Löse folgende Aufgaben (keine Begründung notwendig):

- Für jedes Element der Liste a_i , $0 \leq i \leq 6$, gib die Menge aller möglichen Werte an.
- Die fehlende Zahl kann mit nur einem Quest eindeutig bestimmt werden. Mit welchem Quest und wie kann man die Zahl anschliessend herausfinden?



Teilaufgabe 2: Optimaler Algorithmus (90 Punkte)

Entwickle einen Algorithmus, der das Problem optimal löst. Gegeben n sollst du einige Quests lösen um die fehlende Zahl zu bestimmen. Beachte, dass du deine Strategie abhängig von den Resultaten der Quests anpassen kannst.

Optimiere zuerst nach der Anzahl Quests. Gib eine exakte Formel für die Anzahl nötiger Quests im schlimmsten Fall (z.B. $2^n + n + 13$ oder $\frac{1}{2}n \cdot (n - 1)$). Optimiere dann auf (asymptotische) Laufzeit und dann auf (asymptotische) Speichernutzung.

Binnabike

Maus Binna ist gerade erst in Singapur angekommen für die IOI 2020. Sie ist gerade in ihrem Hotel und möchte jetzt an die Eröffnungszereemonie gehen, und zwar mit dem Velo, da diese ziemlich weit entfernt ist. Dazu mietet Maus Binna ein Binnabike, um durch die Stadt zu fahren. Dies funktioniert so:

- Weil Singapur möchte, dass mehr Leute umweltfreundliche Verkehrsmittel verwenden, hat die Regierung entschieden, dass die ersten d Minuten einer Velofahrt von einer Station aus gratis sind – dies kann mehrmals pro Tag gemacht werden!
- Nach den ersten d Minuten muss für jede weitere Minute ein Singapur-Dollar bezahlt werden.
- Das Binnabike muss immer zu einer offiziellen Station zurückgebracht werden.

In Singapur gibt es n Kreuzungen verbunden durch m Strassen. Für jede dieser Strassen weiss Binna, wie lange sie benötigt; diese Zeit ist gegeben als Zahl > 0 . Sie weiss auch, dass k dieser Kreuzungen offizielle Binnabike-Stationen sind, an denen Velos ausgeliehen und zurückgebracht werden können.

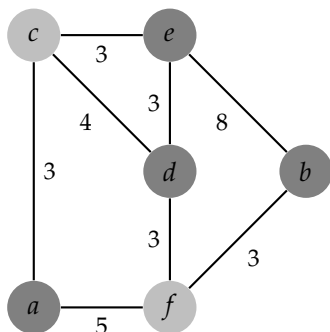
Binna hat herausgefunden, dass Station a am nächsten beim Hotel ist und Station b am nächsten bei der Eröffnungszereemonie. Da die Zeremonie in t Minuten startet, will Binna natürlich innerhalb von t Minuten in Station b ankommen. Trotzdem will Binna so wenig Geld wie möglich für die Fahrt ausgeben. Hilf ihr dabei!

Formal Description Gegeben ist ein Graph mit n Knoten und m Kanten. Die Kanten sind gewichtet mit einer Ganzzahl > 0 . k der Knoten sind Stationen. Es gibt einen Rabatt-Parameter d . Falls Binna in ihrer Route von einer Station u zur nächsten Station v insgesamt w Minuten benötigt, kostet sie dieser Abschnitt $\max(w - d, 0)$. Finde die minimalen Kosten um von Station a nach Station b zu reisen, wobei sie höchstens t Minuten lang unterwegs sein darf.

Es ist garantiert, dass es immer einen Weg von a nach b gibt, der höchstens t lang ist.

Limits Du kannst annehmen, dass n und t sehr gross sind, und d und k dazu vergleichsweise klein. Konkret kannst du annehmen, dass $d \cdot k$ (und damit auch d und k) kleiner sind als t , und dass k^3 kleiner ist als n . Die Kantengewichte sind Ganzzahlen > 0 und können beliebig gross werden.

Teilaufgabe 1: Ein Beispiel lösen (5 Punkte)



Gib den Pfad an, den Binna in dem Graph oben nehmen soll, um die Kosten zu minimieren, mit $d = 3$ und $t = 14$. Die $k = 4$ Stationen sind a, b, d, e . Schreibe die Anzahl Singapur-Dollars auf, die sie bezahlen muss. Es ist keine Begründung notwendig.



Teilaufgabe 2: Unbegrenzte Zeit (35 Punkte)

Weil Binna denkt, dass die Eröffnungszeremonie sowieso jedes Jahr etwa gleich ist, ist es ihr egal ob sie zu spät kommt. Konstruiere einen Algorithmus, der die Kosten minimiert, wenn Binna unbegrenzt Zeit hat.

Teilaufgabe 3: Allgemeiner Fall (60 Punkte)

Dummerweise haben die Schweizer Leiter angeordnet, dass Binna sie bei der Eröffnungszeremonie treffen soll, bevor diese beginnt, das heisst, Binna muss sogar noch früher als erwartet ankommen. Es gibt keine Einschränkungen mehr. Beschreibe deinen Algorithmus.

Venedig

Leider steht Venedig wegen der Meeresspiegelerhöhung unter Wasser und kann nicht mehr besucht werden. Da Maus Stoffl Venedig schon immer einmal besuchen wollte aber nie die Gelegenheit dazu hatte ist er nun zutiefst traurig. Maus Binna hat daher angefangen, die Stadt nachzubauen. Sie hat schon n Sehenswürdigkeiten gebaut und Kanäle zwischen den Sehenswürdigkeiten gegraben. Da es sehr viel Aufwand ist, diese Kanäle zu graben, hat sie gerade so viele gegraben, damit alle Sehenswürdigkeiten durch Kanäle miteinander verbunden sind, aber nicht mehr.

Das wichtigste in Venedig sind die Gondeln: Jede Gondel kann zwischen zwei Sehenswürdigkeiten hin und her fahren und benutzt dazu den direkten Weg, wobei sie bei allen Sehenswürdigkeiten, die auf dem Weg liegen, hält.

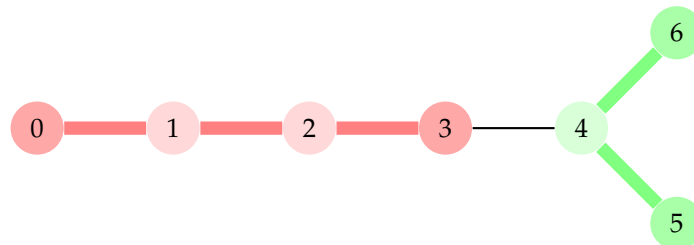
Gondeln zu bauen ist leider auch nicht ganz so einfach. Maus Binna möchte daher die Routen so setzen, dass es möglichst wenige Gondeln braucht, aber an jeder Sehenswürdigkeit mindestens eine Gondel hält. Hilf ihr, herauszufinden, wie viele Gondeln sie benötigt und wie sie die Routen setzen muss.

Formale Beschreibung Gegeben ist der Graph der Sehenswürdigkeiten und die Kanäle dazwischen. Es ist garantiert, dass der Graph ein Baum ist. Insbesondere bedeutet dies, dass es für je zwei Sehenswürdigkeiten a und b genau einen Pfad von a nach b gibt.

Eine Gondelteilung ist eine Liste von Paaren (a_i, b_i) . Dies bedeutet, dass Gondel i zwischen Sehenswürdigkeit a_i und Sehenswürdigkeit b_i hin und her fährt. Eine Gondelteilung heisst valid, falls an jeder Haltestelle mindestens eine Gondel hält. Eine Gondelteilung heisst optimal, falls sie valid ist, und es keine valide Gondelteilung gibt, die weniger Gondeln benötigt.

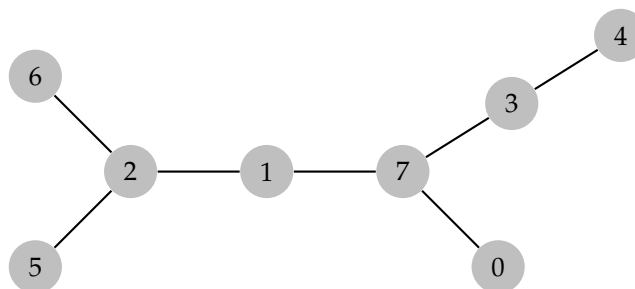
Beispiel $n = 7$ und die Liste der Kanäle ist $\{(0, 1), (1, 2), (2, 3), (3, 4), (4, 5), (4, 6)\}$. Dann ist eine Lösung, eine Gondel zwischen 0 und 3 zu haben und eine weitere Gondel zwischen 5 und 6.

Eine optimale Gondelteilung dieses Beispiels ist unten angezeigt:



Teilaufgabe 1: Löse ein Beispiel (10 Punkte)

In Venedig hat es 8 Sehenswürdigkeiten, die wie im Bild unten angezeigt durch Kanäle verbunden sind. Finde eine optimale Gondelteilung und argumentiere, weshalb diese optimal ist.





Teilaufgabe 2: Mehr Gondeln (25 Punkte)

Maus Binna ist zufrieden, wenn sie eine Gondelteilung findet, die fast optimal ist. Genauer gesagt, falls optimale Gondelteilungen k Gondeln benötigen, möchte sie eine valide Gondelteilung finden, die höchstens $2k$ Gondeln benötigt. Entwickle einen Algorithmus, der eine solche Gondelteilung findet und argumentiere, weshalb dieser korrekt ist.

Teilaufgabe 3: Optimale Gondellimite (65 Punkte)

Gondeln bauen ist doch nicht so einfach wie Maus Binna gehofft hat. Deshalb ist sie nur mit einer optimalen Gondelteilung zufrieden. Entwickle einen Algorithmus, der eine optimale Gondelteilung findet und argumentiere, weshalb dieser korrekt ist.