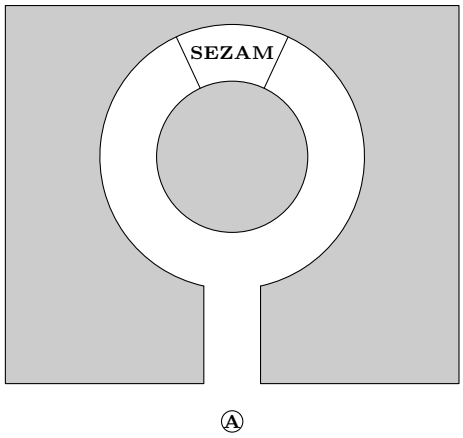
Jak Ali-Baba może nas przekonać, że umie otworzyć Sezam?

*Wojciech GUZICKI**

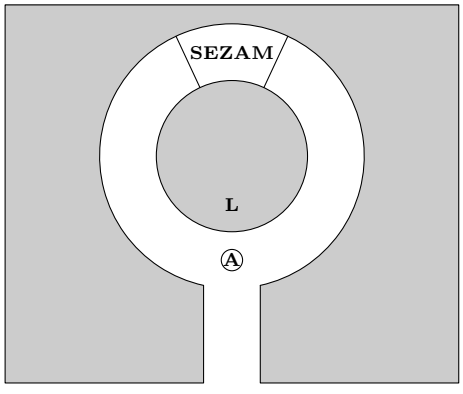
Ali-Baba umie otworzyć Sezam znajdujący się wewnątrz jaskini. Nie chce jednak oczywiście nikomu zdradzić tajnych haseł, które ten Sezam otwierają (znane z literatury hasło „Sezamie, otwórz się" już przestało działać). My nie wierzymy, że Ali-Baba to potrafi. Naturalną metodą przekonania nas, że Ali-Baba umie wejść do Sezamu, jest otworzenie go w naszej obecności. Wtedy jednak moglibyśmy poznać te tajne hasła. Powstaje zatem pytanie o to, w jaki sposób Ali-Baba może nas przekonać, że rzeczywiście umie otworzyć Sezam, nie zdradzając przy tym haseł. Pokażę tutaj, że jest to możliwe. Opiszę procedurę, za pomocą której Ali-Baba będzie mógł przekonać osobę z zewnątrz (tzw. Weryfikatora), że potrafi dostać się do Sezamu. Będzie także widoczne, że w czasie przeprowadzania tej procedury Weryfikator nie pozna haseł otwierających Sezam. Zaczniemy nasze rozważania od przyjrzenia się Sezamowi i zobaczenia, w jaki sposób można wejść do niego.

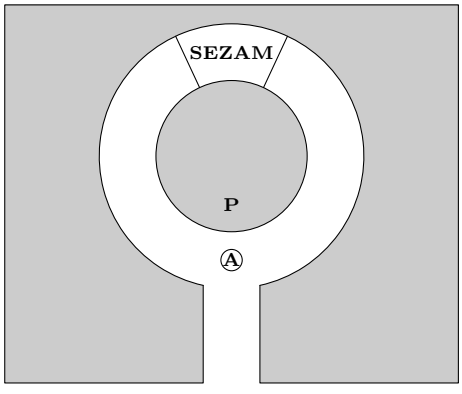
1. Jak Ali-Baba otwiera Sezam?

Ali-Baba przychodzi przed wejście do jaskini, w której znajduje się Sezam.



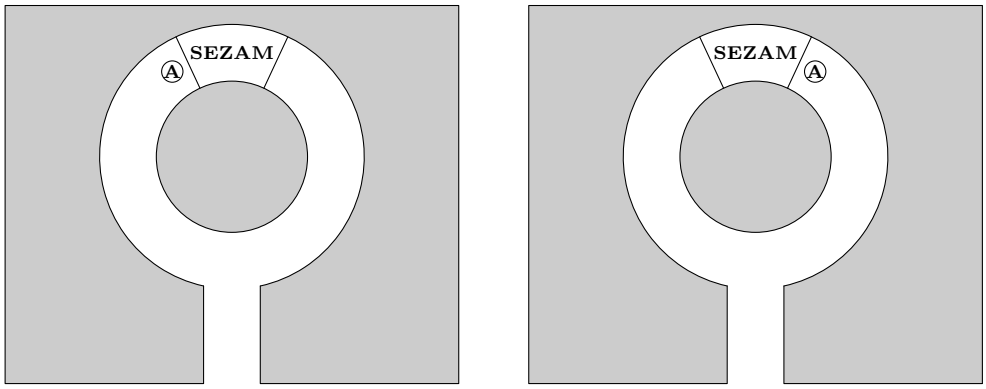
Następnie Ali-Baba wchodzi do jaskini. Wtedy, po usłyszeniu pierwszego hasła, Sezam zdradza mu tajemnicę: czy do Sezamu można wejść przez lewe, czy przez prawe drzwi.



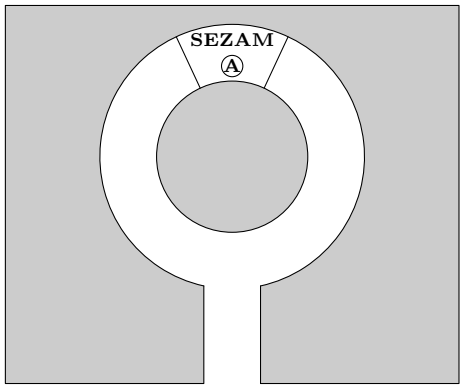


*Wydział Matematyki, Informatyki i Mechaniki, Instytut Matematyki, UW, Banacha 2, 02-097 Warszawa, w.guzicki@mimuw.edu.pl

Teraz Ali-Baba kieruje się we właściwą stronę i dochodzi do drzwi Sezamu.



Wreszcie Ali-Baba otwiera za pomocą drugiego hasła właściwe drzwi i wchodzi do Sezamu.

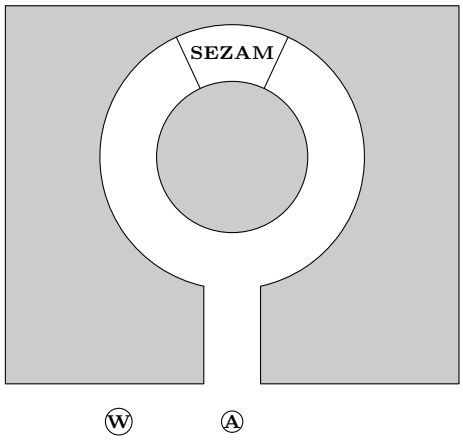


W drodze powrotnej Ali-Baba może wyjść z Sezamu zarówno przez lewe jak i przez prawe drzwi i bez przeszkód wyjść z jaskini.

2. Jak Ali-Baba przekonuje Weryfikatora, że umie otworzyć Sezam?

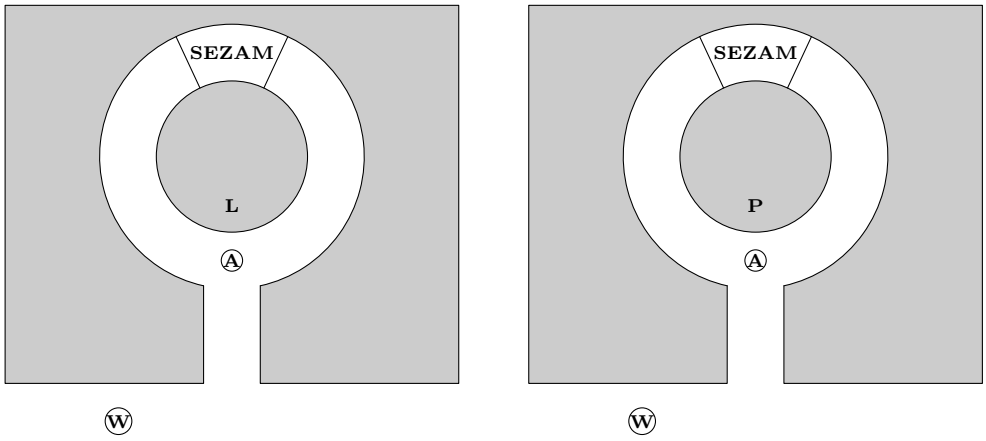
Teraz opiszę procedurę, za pomocą której Ali-Baba przekonuje Weryfikatora, że potrafi wejść do Sezamu. Ta procedura składa się z czterech kroków.

Krok 1. Ali-Baba przychodzi przed wejście do jaskini, w której znajduje się Sezam. Weryfikator czeka z boku i nie widzi wnętrza jaskini.

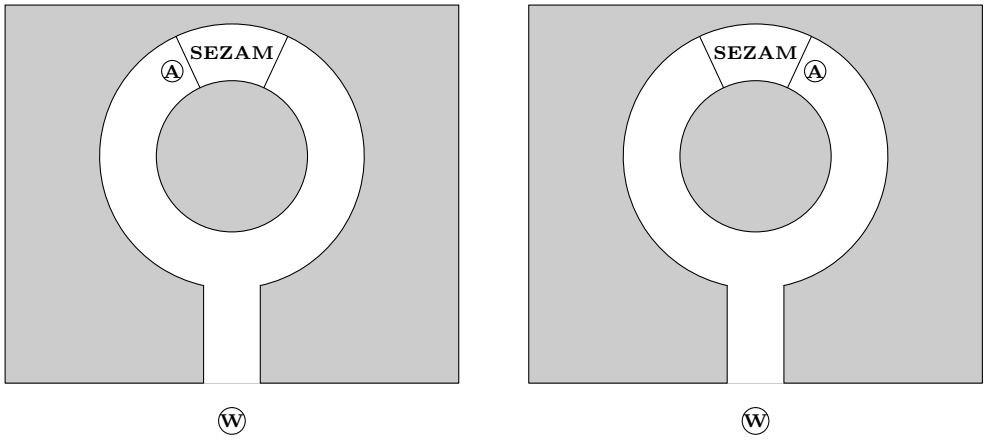


Krok 2. Ali-Baba wchodzi do jaskini i wypowiada pierwsze hasło. Wtedy Sezam zdradza mu tajemnicę: czy do Sezamu można wejść przez lewe czy przez prawe drzwi. Weryfikator nadal czeka z boku. Nie słyszy hasła wypowiadanego przez Ali-Babę i nie widzi wnętrza jaskini. Nie zna zatem odpowiedzi, jakiej udzielił

Sezam Ali-Babie.

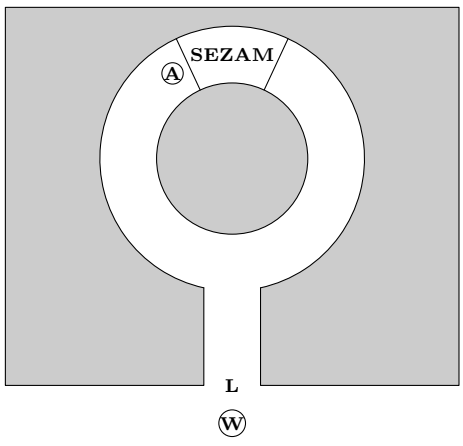


Krok 3. Ali-Baba kieruje się we właściwą stronę i dochodzi do drzwi Sezamu. Weryfikator podchodzi wtedy przed wejście do jaskini. Weryfikator nie widzi już Ali-Baby i oczywiście nie wie, które drzwi dają się otworzyć. Napis, który o tym informował Ali-Babę, już nie jest widoczny.



Krok 4. Weryfikator wybiera jedną z dwóch możliwości: nakazuje Ali-Babie wyjść z jaskini z lewej lub z prawej strony. Ali-Baba słyszy polecenie Weryfikatora. Ali-Baba wykonuje polecenie Weryfikatora. W tym kroku możliwe są cztery przypadki.

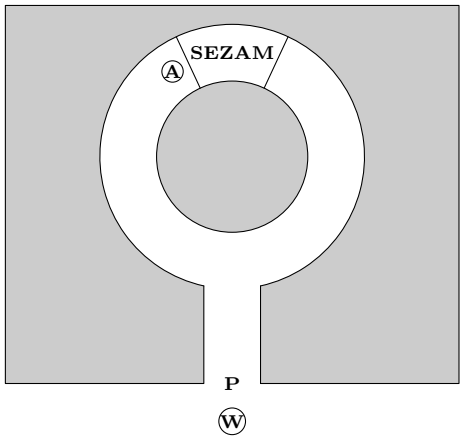
Przypadek 1. Ali-Baba poszedł w lewo i Weryfikator nakazuje mu wrócić z lewej strony.



Ali-Baba wraca do wyjścia i Weryfikator widzi, że Ali-Baba rzeczywiście wrócił z lewej strony.

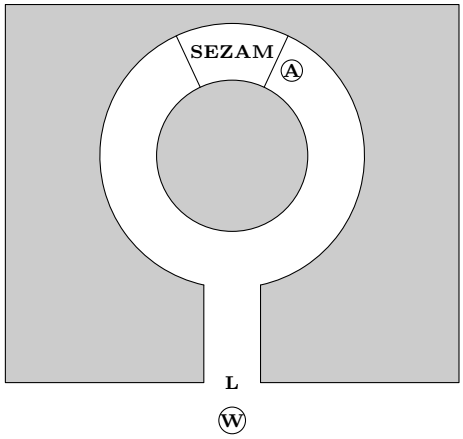
Przypadek 2. Ali-Baba poszedł w lewo i Weryfikator nakazuje mu wrócić

z prawej strony.



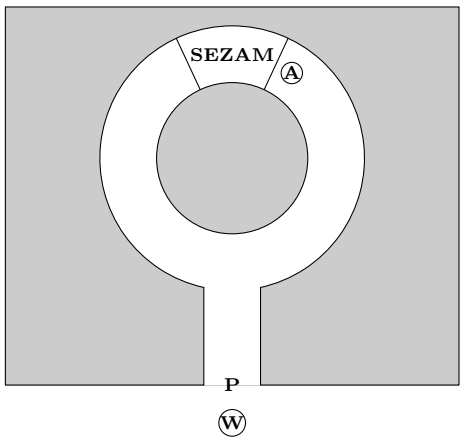
Ali-Baba otwiera Sezam (wypowiadając tajemnicze zaklęcie tak cicho, by Weryfikator tego nie usłyszał), wchodzi do środka, otwiera prawe drzwi i wraca do wyjścia. Weryfikator widzi, że Ali-Baba rzeczywiście wrócił z prawej strony.

Przypadek 3. Ali-Baba poszedł w prawo i Weryfikator nakazuje mu wrócić z lewej strony.



Ali-Baba otwiera Sezam (wypowiadając tajemnicze zaklęcie tak cicho, by Weryfikator tego nie usłyszał), wchodzi do środka, otwiera lewe drzwi i wraca do wyjścia. Weryfikator widzi, że Ali-Baba rzeczywiście wrócił z lewej strony.

Przypadek 4. Ali-Baba poszedł w prawo i Weryfikator nakazuje mu wrócić z prawej strony.



Ali-Baba wraca do wyjścia i Weryfikator widzi, że Ali-Baba rzeczywiście wrócił z prawej strony.

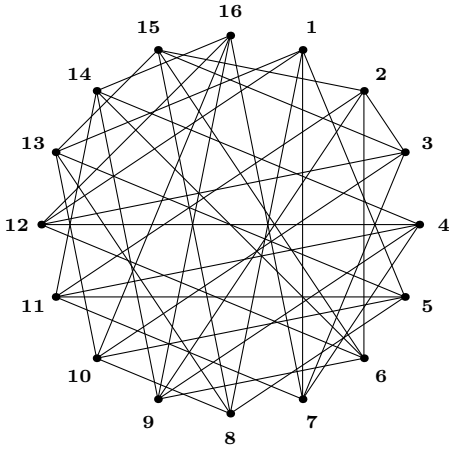
Widzimy, że we wszystkich przypadkach Ali-Baba może wrócić z tej strony, którą wybrał Weryfikator. Teraz tę procedurę Ali-Baba i Weryfikator powtarzają wiele razy. Jeśli za każdym razem Ali-Baba wróci z wybranej przez Weryfikatora

strony, to Weryfikator zostanie przekonany, że Ali-Baba rzeczywiście potrafił przejść przez Sezam, a więc, że rzeczywiście potrafi go otworzyć. Zauważmy bowiem, że mniej więcej w połowie przypadków Weryfikator nakaże Ali-Babie powrócić z innej strony, niż ta, w którą Ali-Baba się udał.

Możemy oczywiście zapytać, dlaczego Weryfikator nie każe po prostu Ali-Babie za każdym razem wyjść z Sezamu na przykład z prawej strony. Przecież w wielu przypadkach Sezam trzeba będzie otworzyć z lewej strony. Ali-Baba pójdzie więc w lewo i będzie musiał przejść przez Sezam, by wyjść z prawej strony. Otóż może się wtedy okazać, że Ali-Baba odgadnie tę strategię Weryfikatora. Gdyby Ali-Baba przewidział, że zostanie za każdym razem poproszony o wyjście z prawej strony, to mógłby zignorować polecenie Sezamu i za każdym razem pójść w prawo. Wtedy będzie mógł spełnić polecenie Weryfikatora niezależnie od tego, czy rzeczywiście potrafi otworzyć Sezam. Ogólnie, gdyby Ali-Baba umiał za każdym razem odgadnąć polecenie Weryfikatora, to za każdym razem mógłby pójść w odpowiednią stronę i wyjść z tej strony bez konieczności otwierania Sezamu. Aby się przed taką ewentualnością zabezpieczyć, Weryfikator losowo wybiera swoje polecenie. Na przykład rzuca monetą i jeśli wypadnie orzeł, każe Ali-Babie wyjść z lewej strony; w przeciwnym przypadku każe mu wyjść z prawej strony. Teraz mniej więcej w połowie przypadków Ali-Baba nie odgadnie polecenia Weryfikatora i będzie musiał otworzyć Sezam, by móc wyjść z właściwej strony. To przekona Weryfikatora, że Ali-Baba rzeczywiście umie otworzyć Sezam.

3. Grafy i izomorfizm grafów

Nie będę tu podawał ścisłej definicji grafu. Ograniczę się do opisu i jednego przykładu. Graf składa się z **wierzchołków** (na rysunku poniżej są to grube kropki) i **krawędzi** (na rysunku są to odcinki łączące wierzchołki). Punkty przecięcia krawędzi nie są wierzchołkami. Opis grafu składa się z listy wierzchołków; przy każdym wierzchołku wymienione są wszystkie wierzchołki, z którymi jest on połączony krawędzią. Oto przykład takiego grafu G mającego 16 wierzchołków i 40 krawędzi wraz z opisem tego grafu. Zauważmy, że z każdego wierzchołka grafu G wychodzi dokładnie 5 krawędzi.



1 : 5, 7, 8, 12, 13,
2 : 3, 6, 9, 11, 15,
3 : 2, 7, 10, 12, 15,
4 : 7, 9, 11, 12, 14,
5 : 1, 8, 10, 11, 13,
6 : 2, 9, 12, 14, 15,
7 : 1, 3, 4, 11, 16,
8 : 1, 5, 10, 13, 15,
9 : 2, 4, 6, 14, 16,
10 : 3, 5, 8, 13, 16,
11 : 2, 4, 5, 7, 14,
12 : 1, 3, 4, 6, 16,
13 : 1, 5, 8, 10, 15,
14 : 4, 6, 9, 11, 16,
15 : 2, 3, 6, 8, 13,
16 : 7, 9, 10, 12, 14.

Przenumerujmy teraz wierzchołki grafu G. Zamiast numeru w górnym wierszu napiszmy numer z dolnego wiersza następującej tabeli:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	3	11	7	13	14	9	2	10	16	6	12	1	4	8	15

Ten sposób zamiany liczb jest pewnym przekształceniem zbioru liczb od 1 do 16 w ten sam zbiór. Oznaczmy je literą f. W przekształceniu f wymagamy, by w dolnym wierszu wystąpiły wszystkie liczby od 1 do 16. Oczywiście wówczas każda liczba wystąpi dokładnie jeden raz. Takie przekształcenie f jest więc **permutacją** liczb od 1 do 16. Możemy zatem napisać inaczej, że:

$$f(1) = 5, \quad f(2) = 3, \quad f(3) = 11, \quad f(4) = 7, \quad f(5) = 13 \text{ i tak dalej.}$$

Teraz w opisie grafu zamieńmy wszędzie numery zgodnie z przekształceniem f (to znaczy, że zamiast numeru i piszemy we wszystkich miejscach numer $f(i)$):

$5:$	$13, 9, 2, 12, 1,$	$10:$	$3, 7, 14, 4, 15,$
$3:$	$11, 14, 10, 6, 8,$	$16:$	$11, 13, 2, 1, 15,$
$11:$	$3, 9, 16, 12, 8,$	$6:$	$3, 7, 13, 9, 4,$
$7:$	$9, 10, 6, 12, 4,$	$12:$	$5, 11, 7, 14, 15,$
$13:$	$5, 2, 16, 6, 1,$	$1:$	$5, 13, 2, 16, 8,$
$14:$	$3, 10, 12, 4, 8,$	$4:$	$7, 14, 10, 6, 15,$
$9:$	$5, 11, 7, 6, 15,$	$8:$	$3, 11, 14, 2, 1,$
$2:$	$5, 13, 16, 1, 8,$	$15:$	$9, 10, 16, 12, 4.$

Następnie uporządkujmy nowe numery:

$1:$	$2, 5, 8, 13, 16,$	$9:$	$5, 6, 7, 11, 15,$
$2:$	$1, 5, 8, 13, 16,$	$10:$	$3, 4, 7, 14, 15,$
$3:$	$6, 8, 10, 11, 14,$	$11:$	$3, 8, 9, 12, 16,$
$4:$	$6, 7, 10, 14, 15,$	$12:$	$5, 7, 11, 14, 15,$
$5:$	$1, 2, 9, 12, 13,$	$13:$	$1, 2, 5, 6, 16,$
$6:$	$3, 4, 7, 9, 13,$	$14:$	$3, 4, 8, 10, 12,$
$7:$	$4, 6, 9, 10, 12,$	$15:$	$4, 9, 10, 12, 16,$
$8:$	$1, 2, 3, 11, 14,$	$16:$	$1, 2, 11, 13, 15.$

Otrzymaliśmy w ten sposób opis nowego grafu H. Jeśli wierzchołek o numerze m był w grafie G połączony krawędziami z wierzchołkami k_1, k_2, k_3, k_4 i k_5, to wierzchołek o numerze $f(m)$ w grafie H będzie połączony z wierzchołkami o numerach $f(k_1), f(k_2), f(k_3), f(k_4)$ i $f(k_5)$. Na przykład, wierzchołek o numerze 7 był w grafie G połączony z wierzchołkami 1, 3, 4, 11 i 16. Mamy

$$f(7) = 9, \quad f(1) = 5, \quad f(3) = 11, \quad f(4) = 7, \quad f(11) = 6, \quad f(16) = 15.$$

Zatem w grafie H wierzchołek o numerze 9 jest połączony krawędziami z wierzchołkami o numerach 5, 11, 7, 6 i 15.

Graf H powstał z grafu G za pomocą przekształcenia f; naturalne jest więc oznaczenie go symbolem $f(G)$. Mamy zatem

$$H = f(G).$$

Przekształcenie f nazywamy **izomorfizmem** grafów. To, że graf H powstał z grafu G za pomocą izomorfizmu f, zapisujemy w następujący sposób:

$$f : G \ \cong> \ H.$$

Przypomnijmy przekształcenie f:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	3	11	7	13	14	9	2	10	16	6	12	1	4	8	15

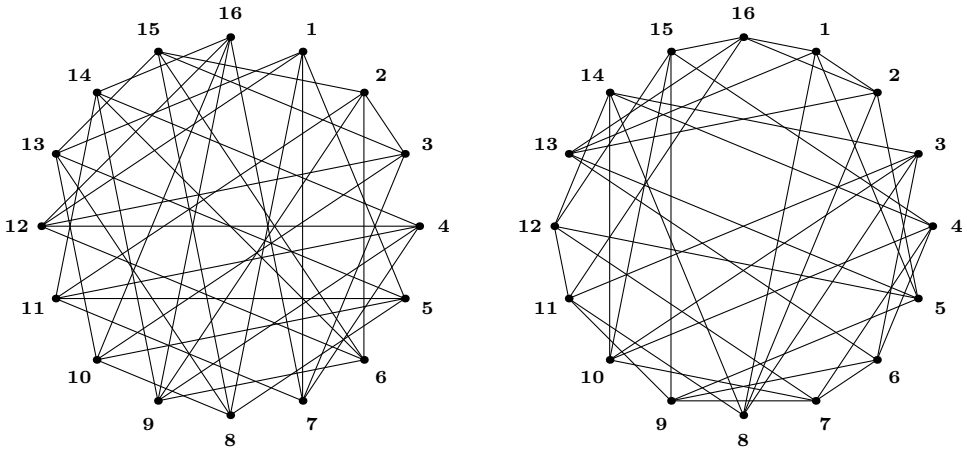
Przekształcenie odwrotne wygląda następująco:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	8	2	14	1	11	4	15	7	9	3	12	5	6	16	10

Jeśli w przekształceniu f liczbę w górnym wierszu zastępowaliśmy liczbą z dolnego wiersza, to w przekształceniu f^{-1} liczby z dolnego wiersza tabeli opisującej przekształcenie f zastępujemy liczbami z górnego wiersza. Oczywiście przekształcenie odwrotne f^{-1} też jest izomorfizmem grafów i mamy

$$f^{-1} : H \ \cong> \ G.$$

Narysujmy teraz oba grafy G i H obok siebie:



Widzimy na pierwszy rzut oka, że te grafy bardzo się od siebie różnią — mimo, iż jeden z nich jest izomorficzną kopią drugiego. Przypuśćmy teraz, że mamy dane dwa dowolne grafy G i H i chcemy dowiedzieć się, czy są one izomorficzne. W niektórych przypadkach możemy łatwo stwierdzić, że izomorficzne nie są. Na przykład wtedy, gdy mają różne liczby wierzchołków lub różne liczby krawędzi. Inny przykład, to sytuacja, w której jeden graf ma wierzchołek, z którego wychodzi 37 krawędzi, a drugi graf takiego wierzchołka nie ma.

Odrzućmy jednak takie sytuacje oczywiste. Przypuśćmy na przykład, że mamy dane dwa grafy o tej samej ogromnej liczbie wierzchołków i o tej własności, że z każdego wierzchołka obu grafów wychodzi ta sama liczba krawędzi. Okazuje się teraz, że pytanie o to, czy takie grafy są izomorficzne, jest bardzo trudne. Nie znamy żadnej **efektywnej** metody stwierdzenia, czy taki izomorfizm grafów istnieje. Nie znamy także żadnej **efektywnej** metody znajdowania takiego izomorfizmu nawet wówczas, gdy wiemy, że te grafy są izomorficzne.

4. Jak Ali-Baba przekonuje Weryfikatora, że zna izomofrizm dwóch grafów?

Nasz Ali-Baba nauczył się w międzyczasie teorii grafów i skonstruował dwa ogromne grafy izomorficzne G i H. Dokładniej: skonstruował graf G mający ogromną liczbę n (na przykład $n = 10^9$) wierzchołków (i dla pewności taki, w którym z każdego wierzchołka wychodzi ta sama liczba k krawędzi — na przykład $k = 1000$). Następnie skonstruował permutację f liczb od 1 do n i stworzył graf $H = f(G)$. Opisy obu grafów zapisał na swojej stronie internetowej i każdy może je obejrzeć. Wreszcie Ali-Baba napisał, że oba te grafy są izomorficzne i on zna taki izomorfizm. Czy może nas o tym przekonać, nie zdradzając przy tym, jak ten izomorfizm wygląda?

Weryfikator chce sprawdzić, czy Ali-Baba mówi prawdę. Procedura, która ma przekonać Weryfikatora, składa się z trzech kroków.

Krok 1. Ali-Baba konstruuje nową permutację g liczb od 1 do n i za pomocą tej permutacji tworzy graf K z grafu G lub z grafu H:

$$K = g(G) \qquad \text{lub} \qquad K = g(H).$$

Mamy zatem

$$g : G \ \cong> \ K \qquad \text{lub} \qquad g : H \ \cong> \ K.$$

Krok 2. Weryfikator prosi Ali-Babę o pokazanie, że nowy graf K jest izomorficzny z grafem G lub z grafem H. Dokładniej, prosi o pokazanie jednego z dwóch izomorfizmów:

$$\varphi : G \ \cong> \ K \qquad \text{lub} \qquad \psi : H \ \cong> \ K.$$

Krok 3. Ali-Baba pokazuje Weryfikatorowi właściwy izomorfizm. Mamy jednak w tym kroku 4 przypadki.

Przypadek 1. Ali-Baba skonstruował graf $K = g(G)$ i Weryfikator prosi go o podanie izomorfizmu grafów $\varphi : G \ \cong> \ K$. Wówczas Ali-Baba pokazuje Weryfikatorowi izomorfizm g.

Przypadek 2. Ali-Baba skonstruował graf $K = g(G)$ i Weryfikator prosi go o podanie izomorfizmu grafów $\psi : H \cong> K$. Wówczas Ali-Baba pokazuje Weryfikatorowi izomorfizm $g \circ f$.

Przypadek 3. Ali-Baba skonstruował graf $K = g(H)$ i Weryfikator prosi go o podanie izomorfizmu grafów $\varphi : G \cong> K$. Wówczas Ali-Baba pokazuje Weryfikatorowi izomorfizm $g \circ f^{-1}$.

Przypadek 4. Ali-Baba skonstruował graf $K = g(H)$ i Weryfikator prosi go o podanie izomorfizmu grafów $\psi : H \cong> K$. Wówczas Ali-Baba pokazuje Weryfikatorowi izomorfizm g.

Tę procedurę Ali-Baba i Weryfikator powtarzają wiele razy, przy czym za każdym razem Ali-Baba wybiera nową permutację g, czyli nowy graf K. Zauważmy, że jeśli za każdym razem Ali-Baba potrafi wskazać żądany izomorfizm, to musiał znać izomorfizm f.

Zauważmy, że gdyby tak jak w przypadku otwierania Sezamu, Ali-Baba umiał przewidzieć, o co zapyta go Weryfikator, to umiałby go oszukać. Na początku Ali-Baba wybrałby dwa dowolne grafy G i H mające n wierzchołków (oraz takie, że z każdego wierzchołka wychodzi taka sama liczba krawędzi). Te grafy nie musiałyby nawet być izomorficzne. Następnie przypuśćmy, że za każdym razem Ali-Baba trafnie odgaduje pytanie Weryfikatora. Wówczas za każdym razem Ali-Baba wybiera dowolną permutację g liczb od 1 do n oraz:

- jeśli Ali-Baba odgadnie, że zostanie poproszony o pokazanie izomorfizmu $\varphi : G \cong> K$, to tworzy graf $K = g(G)$,
- jeśli natomiast Ali-Baba odgadnie, że zostanie poproszony o pokazanie izomorfizmu $\psi : H \cong> K$, to tworzy graf $K = g(H)$.

Wówczas oczywiście Ali-Baba spełni żądanie Weryfikatora. Aby przed takim oszustwem się zabezpieczyć, Weryfikator **losowo** wybiera pytanie o izomorfizm $\varphi : G \cong> K$ lub $\psi : H \cong> K$. Na przykład rzuca monetą i jeśli wypadnie orzeł, to prosi o izomorfizm φ, a jeśli wypadnie reszka, to prosi o izomorfizm ψ. Teraz Ali-Baba nie może oszukać Weryfikatora. Jeśli bowiem będzie próbował odgadnąć prośbę Weryfikatora, to mniej więcej w połowie przypadków nie odgadnie i wtedy:

- albo Ali-Baba przyjmie, że zostanie poproszony o pokazanie izomorfizmu $\varphi : G \cong> K$, utworzy graf $K = g(G)$, ale Weryfikator poprosi go o pokazanie izomorfizmu $\psi : H \cong> K$,
- albo Ali-Baba przyjmie, że zostanie poproszony o pokazanie izomorfizmu $\psi : H \cong> K$, utworzy graf $K = g(H)$, ale Weryfikator poprosi go o pokazanie izomorfizmu $\varphi : G \cong> K$.

W obu przypadkach Ali-Baba nie będzie umiał wskazać właściwego izomorfizmu, gdyż nie zna potrzebnego w tych przypadkach izomorfizmu $f : G \cong> H$. A więc jeśli za każdym razem Ali-Baba spełni wymaganie Weryfikatora, to znaczy, że albo za każdym razem trafnie odgadł jego pytanie (co przy dużej liczbie pytań jest bardzo nieprawdopodobne), albo rzeczywiście znał izomorfizm grafów G i H.

5. Kwadraty modulo n

Nasze rozważania rozpoczniemy od wybrania dwóch dużych liczb pierwszych p i q. Nie będziemy tutaj zajmowali się tym, w jaki sposób możemy znaleźć takie liczby pierwsze. Czytelnikowi powinna wystarczyć informacja, że znane są efektywne metody znajdowania liczb pierwszych mających kilkaset cyfr. Teraz obliczamy iloczyn tych dwóch liczb pierwszych:

$$n = p \cdot q.$$

Następnie wybieramy dowolną liczbę całkowitą x spełniającą nierówności

$$1 \leqslant x \leqslant n - 1$$

i podnosimy ją do kwadratu modulo n:

$$y = x^2 \ \mathbf{mod}\ n.$$

W tym miejscu należy się krótkie wyjaśnienie. Oznaczenie a **mod** b oznacza resztę z dzielenia a przez b. Zatem liczbę y obliczamy podnosząc x do kwadratu,

następnie dzieląc otrzymany kwadrat przez n i wreszcie biorąc jako wynik otrzymaną resztę z dzielenia. Liczbę y nazywamy **kwadratem** liczby x modulo n. Liczba y spełnia nierówności

$$1 \leqslant y \leqslant n-1.$$

To, że $y \leqslant n-1$, jest oczywiste: liczba y jest przecież resztą z dzielenia przez n. Musimy tylko wykazać, że $y \neq 0$. Przypuśćmy zatem, że $y = 0$. To znaczy, że liczba x^2 jest podzielna przez n, czyli

$$p \mid x^2 \quad \text{oraz} \quad q \mid x^2.$$

Ponieważ p i q są liczbami pierwszymi, więc także

$$p \mid x \quad \text{oraz} \quad q \mid x.$$

Stąd jednak wynika, że

$$pq \mid x,$$

czyli

$$n \mid x.$$

To jednak jest niemożliwe, gdyż $x \leqslant n-1$.

Teraz interesuje nas działanie odwrotne. Przypuśćmy, że mamy daną dowolną liczbę całkowitą y spełniającą nierówności

$$1 \leqslant y \leqslant n-1$$

i chcemy znaleźć liczbę całkowitą x taką, że

$$y = x^2 \ \mathbf{mod}\ n.$$

Przede wszystkim okazuje się, że nie zawsze taka liczba x istnieje. Można bowiem udowodnić, że jeśli liczba całkowita y jest kwadratem pewnej liczby x modulo n, to istnieją dokładnie cztery takie liczby x. Oczywiście każdą z tych czterech liczb x nazywamy **pierwiastkiem** z liczby y modulo n. Ponadto dla różnych liczb y (będących kwadratami modulo n) odpowiednie zbiory czterech pierwiastków są rozłączne. Stąd wynika, że wśród liczb od 1 do $n-1$ istnieje dokładnie $\frac{n-1}{4}$ liczb będących kwadratami modulo n. Przypuśćmy następnie, że liczba y jest takim kwadratem i chcemy znaleźć którykolwiek pierwiastek z liczby y modulo n, czyli którąkolwiek liczbę całkowitą x taką, że

$$y = x^2 \ \mathbf{mod}\ n.$$

Okazuje się wówczas, że jest to możliwe tylko wtedy, gdy znamy obie liczby pierwsze p i q. Mianowicie

- nie jest znana żadna **efektywna** metoda znajdowania któregokolwiek pierwiastka z y modulo n, jeśli dane są wyłącznie liczby n i y,
- jest znana **efektywna** metoda znajdowania wszystkich czterech pierwiastków z y modulo n, jeśli dane są liczby p, q i y.

Co więcej, gdybyśmy znali wyłącznie n i y oraz poznali w jakikowiek sposób wszystkie cztery pierwiastki z y modulo n, to moglibyśmy **efektywnie** obliczyć liczby p i q.

Widzimy więc, że znajdowanie pierwiastków modulo n jest tak samo trudne, jak rozkładanie dużych liczb złożonych na czynniki pierwsze. Obecnie najlepsze znane algorytmy faktoryzacji (czyli rozkładania na czynniki) uruchamiane równolegle na wielu komputerach pozwalają rozkładać na czynniki liczby mające około 200 cyfr. Jeśli zatem wybierzemy liczby pierwsze p i q mające po około 200 cyfr, to rozłożenie na czynniki liczby n (mającej około 400 cyfr) będzie w praktyce niewykonalne.

Znajomość jednego pierwiastka nie wystarczy do rozłożenia liczby n na czynniki. Możemy bowiem wybrać dowolną liczbę x i obliczyć jej kwadrat y. Wówczas oczywiście znamy jeden pierwiastek z y modulo n: jest nim wybrana na początku liczba x (znamy nawet dwa pierwiastki, drugim jest $n-x$). Nie umiemy natomiast obliczyć pozostałych pierwiastków.

6. Jak Ali-Baba przekonuje Weryfikatora, że zna pierwiastek kwadratowy modulo n?

Przypuśćmy teraz, że dana jest duża liczba n będąca iloczynem dwóch dużych liczb pierwszych p i q. Taką liczbę Ali-Baba może obliczyć sam lub też może skorzystać z tzw. Zaufanego Centrum dostarczającego użytkownikom potrzebne liczby. Takie liczby są na przykład wykorzystywane do szyfrowania bardzo popularnym systemem kryptograficznym RSA. Tak więc Ali-Baba posiada znaną publicznie liczbę $n = pq$. Teraz Ali-Baba wybiera losowo liczbę x spełniającą nierówności

$$1 \leqslant x \leqslant n - 1$$

i oblicza jej kwadrat modulo n:

$$y = x^2 \textbf{ mod } n.$$

Liczbę y Ali-Baba ogłasza publicznie i twierdzi, że zna co najmniej jeden pierwiastek kwadratowy z y modulo n. Twierdzi przy tym, że może przekonać dowolnego Weryfikatora, że taki pierwiastek zna, nie ujawniając go przy tym. A jak to robi?

Ali-Baba wraz z Weryfikatorem przeprowadzają wielokrotnie (w praktyce wystarczy około 20 razy) następującą procedurę:

- Ali-Baba wybiera losowo liczbę całkowitą v spełniającą nierówności $1 \leqslant v \leqslant n - 1$, oblicza jej kwadrat $w = v^2$ **mod** n i przesyła ten kwadrat Weryfikatorowi,
- Weryfikator prosi Ali-Babę o jedną z dwóch rzeczy: podanie liczby v lub podanie iloczynu xv **mod** n,
- w pierwszym przypadku Ali-Baba wysyła liczbę v i Weryfikator sprawdza, czy istotnie $w = v^2$ **mod** n,
- w drugim przypadku Ali-Baba wysyła liczbę xv **mod** n i Weryfikator sprawdza, czy zachodzi równość

$$(xv)^2 \textbf{ mod } n = (yw) \textbf{ mod } n.$$

Jeśli za każdym razem Weryfikator stwierdzi, że odpowiednia równość jest prawdziwa, to będzie (z dużym prawdopodobieństwem) przekonany, że istotnie Ali-Baba zna co najmniej jeden pierwiastek kwadratowy z y modulo n. Można bowiem wykazać, że jeśli Ali-Baba nie zna pierwiastka kwadratowego z y modulo n, to nie będzie umiał poprawnie wskazać obu liczb v i xv **mod** n.

7. Po co to wszystko?

Opisane procedury noszą nazwę **dowodów o zerowej wiedzy** (po angielsku: *zero-knowledge proofs*). Polegają one na tym, by przekonać Weryfikatora o tym, że posiada się pewną wiedzę na jakiś temat, nie zdradzając przy tym ani odrobiny tej wiedzy. Za pomocą odpowiednich dowodów o zerowej wiedzy można w ten sposób przekonać o posiadaniu praktycznie dowolnej informacji (której zdobycie przez osobę postronną jest w praktyce niewykonalne), nie zdradzając przy tym żadnej części tej tajnej informacji.

Przykładowym zastosowaniem takich dowodów o zerowej wiedzy jest identyfikacja przy dostępie do jakiegoś systemu informatycznego (na przykład zalogowanie się do swojego konta internetowego w banku). Użytkownik — nazwijmy go w dalszym ciągu Ali-Babą — wybiera pewną tajną informację jako hasło dostępu. Na przykład wybiera dwa ogromne izomorficzne grafy. Same grafy są znane publicznie i nikt nie umie znaleźć izomorfizmu przekształcającego jeden na drugi. Ten izomorfizm, znany wyłącznie Ali-Babie, jest tajną częścią hasła. Teraz Ali-Baba loguje się do systemu. System występuje w roli Weryfikatora i sprawdza za pomocą odpowiedniej liczby testów, czy Ali-Baba rzeczywiście zna ten tajny izomorfizm. Jeśli zostanie o tym przekonany, to umożliwi Ali-Babie dostęp do konta. Czym ta procedura różni się od powszechnie stosowanych haseł? Otóż zauważmy, że Weryfikator (w tym przypadku bank) nie posiada kopii hasła (nawet w postaci zaszyfrowanej) i nie sprawdza, czy podane przez Ali-Babę hasło jest zgodne z posiadaną kopią. Jedynym posiadaczem hasła jest sam Ali-Baba. Weryfikacja jego tożsamości polega na stwierdzeniu, że zna on tajny izomorfizm. A skoro tylko on jeden może go znać, więc osobą logującą się do systemu musi być właśnie on.