Rozwiązanie zadania M 1273.
Z danego równania wynika, że liczba x jest parzysta. Niech więc $x = 2a$.
Po podstawieniu dane równanie przybiera postać: $4a^3 + y^3 = 2z^3$.

Stąd wynika, że liczba y jest parzysta. Przyjmijmy zatem $y = 2b$. Wobec tego $2a^3 + 4b^3 = z^3$. Z zależności tej wynika z kolei, że również liczba z jest parzysta. Podstawmy więc $z = 2c$ do ostatniego równania: $a^3 + 2b^3 = 4c^3$.

Wykazaliśmy zatem, że jeśli trójka liczb całkowitych (x, y, z) spełnia dane równanie, to każda z liczb x, y, z jest parzysta i trójka $(\frac{1}{2}x, \frac{1}{2}y, \frac{1}{2}z)$ także spełnia dane równanie.

Kontynuując to rozumowanie, dochodzimy do wniosku, że jeśli trójka liczb całkowitych (x, y, z) spełnia dane równanie, to każda z liczb x, y, z musi być podzielna przez dowolną potęgę dwójki. To jednak jest możliwe jedynie wtedy, gdy $x = y = z = 0$.

Trójka $(x, y, z) = (0, 0, 0)$ jest zatem jedynym rozwiązaniem w liczbach całkowitych danego równania.

Informatyczny kącik olimpijski (29): Trening

Tym razem przyjrzymy się zadaniu *Training* z Międzynarodowej Olimpiady Informatycznej 2007.

Mamy dany spójny graf nieskierowany o n wierzchołkach z wyróżnionym drzewem rozpinającym (tzn. wyróżnione jest $n-1$ krawędzi, które tworzą drzewo). Krawędzie spoza tego drzewa mają przypisane koszty. Należy przeciąć krawędzie o najmniejszej sumie kosztów, tak aby w grafie nie pozostał żaden cykl prosty (tzn. taki, w którym nie powtarzają się wierzchołki) długości parzystej. Krawędzi drzewowych nie wolno przecinać. Dodatkowo wiadomo, że stopień żadnego z wierzchołków grafu nie przekracza (niedużego) k.

Krawędzie spoza drzewa będę nazywał *poprzecznymi*. Każda poprzeczna krawędź tworzy z naszym drzewem dokładnie jeden cykl (nazwijmy go *cyklem poprzecznym*). Jeśli cykl ten jest długości parzystej, musimy tę krawędź przeciąć, robimy więc to czym prędzej i nie zajmujemy się więcej taką krawędzią. Zauważmy, że w poprawnym rozwiązaniu możemy pozostawić cykle poprzeczne długości nieparzystej, ale tylko pod warunkiem, że żadne dwa z nich nie mają wspólnej krawędzi. Gdyby bowiem miały, to przez ich złączenie i usunięcie części wspólnej uzyskalibyśmy cykl prosty długości $(2a+1) + (2b+1) - 2d = 2(a+b+1-d)$, przy czym $2a+1$ i $2b+1$ są długościami tych cykli, a d – długością części wspólnej. Cykli poprzecznych bez wspólnych krawędzi nie da się w żaden sposób połączyć w cykl prosty. Zadanie sprowadza się więc do wybrania poprzecznych krawędzi, które możemy pozostawić, tak aby odpowiadające im cykle poprzeczne nie miały wspólnych krawędzi.

Do rozwiązania tak uproszczonego problemu użyjemy programowania dynamicznego. Ukorzeńmy nasze drzewo rozpinające. Dla każdego wierzchołka x i dla każdego podzbioru S synów x w naszym drzewie obliczymy $t(x, S)$ – największą możliwą sumę kosztów poprzecznych krawędzi, które możemy pozostawić, a których cykle są rozłączne krawędziowo i zawierają się całkowicie w podgrafie złożonym z krawędzi łączących x z wierzchołkami z S, poddrzew wierzchołków z S oraz poprzecznych krawędzi.

Aby obliczyć $t(x, S)$, musimy zastanowić się nad cyklami poprzecznymi, które przechodzą przez x, ale nie przechodzą przez ojca x. Jeśli S jest pusty, to $t(x, S) = 0$. W przeciwnym przypadku ustalmy pierwszego syna $s \in S$. Załóżmy, że istnieje cykl poprzeczny, który zawiera x i s, ale nie zawiera ojca x. Niech e będzie krawędzią poprzeczną tegoż cyklu, a C – zbiorem jego wierzchołków. Wtedy przy obliczaniu $t(x, S)$ należy uwzględnić wyrażenie:

$$(1) \qquad t(x, S \setminus C) + \sum_{r \in C \setminus \{x\}} t(r, A(r) \setminus C),$$

przy czym $A(v)$ oznacza zbiór wszystkich synów wierzchołka v. Możemy też nie zdecydować się wziąć żadnego takiego cyklu, czemu odpowiada wyrażenie:

$$(2) \qquad t(s, A(s)) + t(x, S \setminus \{s\}).$$

Ostatecznie jako $t(x, S)$ weźmiemy maksimum z wartości (1) po odpowiednich parach (e, C) oraz (2).

Odpowiedzią w naszym zadaniu będzie

$$W - t(q, A(q)),$$

przy czym W jest sumą kosztów wszystkich krawędzi poprzecznych, a q – korzeniem drzewa.

Zastanówmy się nad złożonością czasową tego rozwiązania. Graf ma $O(n \cdot k)$ krawędzi. Dla każdej krawędzi poprzecznej musimy znaleźć jej cykl w czasie $O(n)$, co daje łączny koszt $O(n^2 \cdot k)$. Każda krawędź poprzeczna e jest rozważana podczas programowania dynamicznego tylko dla wierzchołka x będącego najwyższym wierzchołkiem w drzewie, do którego sięga cykl poprzeczny zawierający e. Stąd sumę po wierzchołkach tego cyklu, występującą w wyrażeniu (1), wystarczy obliczyć raz. Dla każdej pary (x, S) wybieramy albo brak krawędzi poprzecznych, albo też jedną z krawędzi, których cykle przechodzą przez x oraz pierwszego syna x w S, ale nie przez ojca x. Niech takich właśnie krawędzi poprzecznych będzie $E(x, S)$, a wszystkich krawędzi, których cykle przechodzą przez x, ale nie przechodzą przez jego ojca – $E(x)$. Wtedy

$$\sum_{x \in V} \sum_{S \subseteq A(x)} E(x, S) \leqslant 2^k \cdot \sum_{x \in V} E(x) = \\ = O(n \cdot k \cdot 2^k).$$

Programowanie dynamiczne zajmuje więc czas $O(n \cdot k \cdot 2^k)$ plus obliczanie sum po wstawianych cyklach, którego koszt czasowy jest taki sam jak koszt szukania tych cykli. Stąd łączna złożoność czasowa to $O(n \cdot k \cdot (n + 2^k))$ przy zużyciu pamięci rzędu $O(n \cdot 2^k)$.

Tomasz KULCZYŃSKI