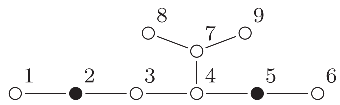
Informatyczny kącik olimpijski (83): Więzienie

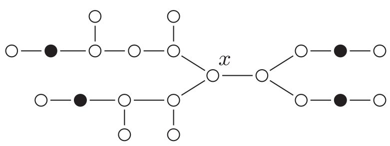
Uczestnicy VII obozu informatycznego ILOCAMP mieli do rozwiązania takie oto zadanie: *Więzienie składa się z n cel i $n-1$ łączących je korytarzy. W części z cel znajdują się więźniowie. Reszta cel jest pusta. W wyniku awarii wszystkie cele zostały otwarte. Jaka jest minimalna liczba strażników potrzebnych do uniemożliwienia więźniom ucieczki? Strażników można rozmieszczać jedynie w pustych celach, a więzień może uciec, jeśli na drodze od jego celi do celi wyjściowej* (*takiej, do której prowadzi tylko jeden korytarz*) *nie ma żadnej celi ze strażnikiem.*



Rys. 1. Przykładowe drzewo T_1 o $n=9$ wierzchołkach i wyróżnionym zbiorze $B=\{2,5\}$. Mamy tu dwa minimalne zbiory blokujące: $\{1,4,6\}$ i $\{1,6,7\}$.

Zadanie można przeformułować następująco: dane jest n-wierzchołkowe drzewo, w którym wyróżniono podzbiór wierzchołków B. Należy znaleźć najmniejszy *blokujący* zbiór wierzchołków (rozłączny z B), taki że każda ścieżka z dowolnego wierzchołka zbioru B do dowolnego liścia drzewa musi przechodzić przez co najmniej jeden wierzchołek ze zbioru blokującego (patrz drzewo T_1 na rysunku 1).

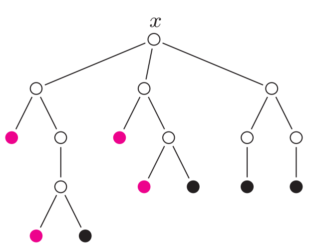
Na początek wypróbujmy kilku kandydatów na zbiór blokujący. Oczywiste jest, że zbiór wszystkich liści w drzewie jest zbiorem blokującym (zakładamy, że B nie zawiera liścia, w przeciwnym przypadku zbiór blokujący nie istnieje). Również jest nim zbiór wszystkich niewyróżnionych sąsiadów wierzchołków z B. Dość łatwo się jednak przekonać, że żaden z nich nie musi być minimalny – dla drzewa T_1 są to zbiory $\{1,6,8,9\}$ oraz $\{1,3,4,6\}$; oba mają po 4 wierzchołki.



Rys. 2. Drzewo T_2. Minimalny zbiór blokujący ma 7 wierzchołków.

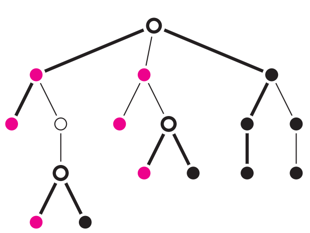
Zacznijmy jednak od zbioru liści i spróbujmy usunąć z niego nadmiarowe wierzchołki. Jeśli jedyny sąsiad u liścia w należy do zbioru B, to każdy zbiór blokujący musi zawierać liść w (np. każdy zbiór blokujący dla drzewa T_1 musi zawierać wierzchołki 1 i 6). Z drugiej strony, jeśli wierzchołek u nie należy do B, to istnieje minimalny zbiór blokujący, który nie zawiera liścia w. Jest tak dlatego, że każda ścieżka z B do w przechodzi przez u, więc, blokując wierzchołek u zamiast wierzchołka w, dostajemy równie dobre rozwiązanie. Postępując w ten sposób z drzewem T_1, pozbędziemy się liści i znajdziemy zbiór blokujący $\{1,4,6\}$, ale nie dla każdego drzewa tak znaleziony zbiór będzie minimalny (patrz drzewo T_2 na rysunku 2).

Zwodnicze jest, że w treści zadania liście oraz wyróżnione wierzchołki pełnią nieco inną rolę (w szczególności można blokować liście, ale nie wierzchołki z B). Możemy jednak nieznacznie przeformułować treść zadania, aby te role stały się w pełni symetryczne. Zauważyliśmy już, że musimy blokować jedynie liście sąsiadujące z B, a ponieważ nie wpływają one na resztę wierzchołków, możemy je dodać do zbioru blokującego i usunąć z drzewa. Następnie oznaczmy zbiór pozostałych liści przez A. *Składową* w drzewie nazwiemy maksymalny zbiór wierzchołków, z których każde dwa można połączyć ścieżką, której wewnętrzne wierzchołki nie należą do zbioru $A \cup B$. Przykładowo w drzewie T_2 mamy jedną składową zawierającą 17 wierzchołków, a gdyby wierzchołek x należał do B, to mielibyśmy trzy składowe, o licznościach 6, 6 i 7. Zauważmy, że zbiór blokujący możemy wyznaczyć dla każdej składowej niezależnie. Będzie to taki podzbiór wierzchołków spoza $A \cup B$, że każda ścieżka ze zbioru A do zbioru B musi przecinać ten podzbiór.



Rys. 3. Jedyna składowa drzewa T_2, ukorzeniona w x. Wierzchołki z A zaznaczone zostały kolorem, wierzchołki z B są czarne.

Weźmy zatem składową i ukorzeńmy ją w dowolnym wierzchołku spoza $A \cup B$ (rys. 3). Zastosujemy teraz programowanie dynamiczne – przeglądając drzewo składowej od liści do korzenia, w każdym poddrzewie będziemy konstruować minimalny zbiór blokujący. Jednocześnie każdy wierzchołek w będziemy wrzucać do jednego z trzech zbiorów: S_A (jeśli poddrzewo zaczepione w w zawiera taki wierzchołek z A, do którego ścieżka z w nie zawiera zablokowanych wierzchołków), S_B (poddrzewo to zawiera niezablokowany wierzchołek z B) oraz S (wszystkie wierzchołki z $A \cup B$ w tym poddrzewie są zablokowane – być może z powodu dodania wierzchołka w do zbioru blokującego).



Rys. 4. Wierzchołki składowej przydzielone do zbiorów S, S_A, S_B zaznaczone są odpowiednio na biało, kolorowo i czarno. Wyróżniono również wierzchołki minimalnego zbioru blokującego oraz odpowiadające im ścieżki.

Wierzchołki ze zbiorów A i B lądują odpowiednio w zbiorach S_A i S_B. Rozważmy teraz wierzchołek wewnętrzny w; jeśli ma on takie dwoje dzieci, z których jedno należy do S_A, a drugie należy do S_B, to przez wierzchołek w przechodzi (co najmniej jedna) niezablokowana ścieżka p_w łącząca pewien wierzchołek z A z pewnym wierzchołkiem z B. Dodajemy zatem w do zbioru blokującego oraz do zbioru S. W przeciwnym przypadku wszystkie dzieci w należą do zbioru S, $S_A \cup S$ lub $S_B \cup S$ i dodajemy wierzchołek w odpowiednio do zbioru S, S_A lub S_B (rys. 4).

Jasne jest, że w ten sposób skonstruujemy poprawny zbiór blokujący W. Dowodem na jego minimalność jest skonstruowany, niejako przy okazji, zbiór ścieżek $\{p_w \mid w \in W\}$, które są rozłączne, a każda musi zawierać co najmniej jeden wierzchołek z W. Złożoność czasowa rozwiązania to $O(n)$.

Tomasz IDZIASZEK