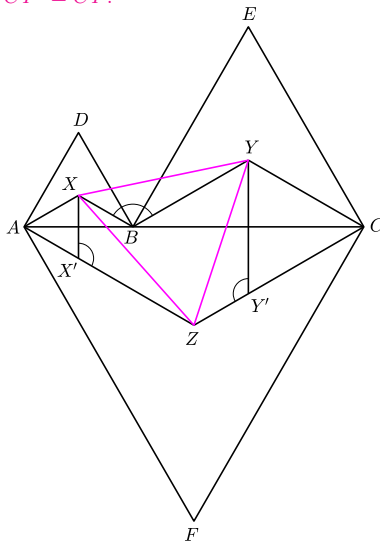


Jeśli mamy $n = 6$ pastwisk jak powyżej (liczba owiec w prostokątach), to jeden z optymalnych planów smoka jest następujący: pożera on 9 owiec z trzeciego pastwiska (płosząc 5 owiec z pierwszych dwóch pastwisk), następnie zjada 6 owiec z piątego pastwiska i w końcu 1 owcę z czwartego pastwiska.



Rozwiązanie zadania M 1432. Niech X' będzie takim punktem na odcinku AZ , że $AX' = AX$ oraz niech Y' będzie takim punktem na odcinku CZ , że $CY' = CY$.



Ponieważ $\sphericalangle XAZ = 60^\circ$, więc trójkąt AXX' jest równoboczny. Podobnie trójkąt YCY' jest równoboczny. Zauważmy, że

$$\sphericalangle XBY = \sphericalangle XX'Z = \sphericalangle YY'Z = 120^\circ.$$

Skoro $AB + BC = CA$, to $AX + CY = AZ$, więc $X'Z = Y'Y = BY$, $ZY' = X'X = XB$. Zatem trójkąty $XB Y$, $XX'Z$, $ZY'Y$ są przystające (cecha bbk). W szczególności $XY = YZ = ZX$.

Informatyczny kącik olimpijski (75): Smok i owce

Tym razem zadanie *Smok* z Akademickich Mistrzostw Polski w Programowaniu Zespołowym 2005, które odbyły się na Uniwersytecie Jagiellońskim. W długiej i głębokiej dolinie znajduje się n kolejno położonych pastwisk, na i -tym z nich pasie się $a[i]$ owiec. Każdego dnia dolinę może nawiedzić smok i pożreć owce z jednego pastwiska. Smok musi zawsze lecieć wzdłuż doliny (ale może wybrać, z której strony przyleci danego dnia), a z każdego pastwiska, nad którym przeleci, wszystkie owce uciekają. Ponadto pod koniec każdego dnia liczebność owiec (czy to z powodu wilków, czy pogłosek o smokach) na każdym pastwisku zmniejsza się o jeden. Należy obliczyć, ile maksymalnie owiec uda się pożreć smokowi (patrz rysunek).

Zauważmy, że niezależnie od poczynań smoka pastwiska, które nie zostały przez niego odwiedzone (tzn. owce na tych pastwiskach nie zostały przez niego ani pożarte, ani spłoszone), leżą w spójnym fragmencie doliny. To sugeruje następujące rozwiązanie dynamiczne: wypełniamy tabelkę, w której $d[i, j, k]$ oznacza maksymalną liczbę pożartych owiec, jeśli pastwiska nieodwiedzone leżą w przedziale $[i, j]$, a smok wykonuje k -ty lot. Pomijając przypadki brzegowe, następująca rekurencja pozwala wyznaczyć tabelkę w czasie $O(n^3)$:

$$d[i, j, k] = \max(d[i + 1, j, k], d[i, j - 1, k], \\ \max(0, a[i] - k + 1) + d[i + 1, j, k + 1], \\ \max(0, a[j] - k + 1) + d[i, j - 1, k + 1]).$$

Widać, że problemy smoka są dwa: które pastwiska wybrać i w jakiej kolejności je odwiedzać. Załóżmy, że w pewnym optymalnym rozwiązaniu smok odwiedza kolejno pastwiska o numerach i_1, i_2, \dots, i_k . Ponieważ na każdym z nich się posili, zatem $a[i_j] \geq j$, i w sumie zje $A = \sum_{j=1}^k (a[i_j] - j + 1)$ owiec. A czy istnieje inna kolejność, w której może on odwiedzać te pastwiska? Zauważmy, że liczba owiec z pastwiska i_j , które nie zostaną pożarte przez smoka, wynosi $j - 1$, zatem w sumie nie pożre on $\binom{k}{2}$ owiec. Ale jeśli smok nie będzie latał kompletnie bez sensu (tzn. nie będzie płoszył owiec na pastwiskach i_1, \dots, i_k), to po j -tym przelocie liczba owiec zmniejszy się co najwyżej o $k - j$, zatem w sumie zmniejszy się co najwyżej o $\binom{k}{2}$. Zatem, odwiedzając pastwiska np. w kolejności od prawej do lewej, smok zje co najmniej A owiec, czyli tyle, ile w rozwiązaniu optymalnym.

To pozwala nam uprościć i przyspieszyć nasze rozwiązanie dynamiczne do działającego w czasie $O(n^2)$, rozważając jedynie przedziały postaci $[1, j]$:

$$d[j, k] = \max(d[j - 1, k], \max(0, a[j] - k + 1) + d[j - 1, k + 1]).$$

Spróbujmy podejść do problemu jeszcze inaczej. Załóżmy przez chwilę, że smok swoim lotem nie straszy owiec. Może wtedy odwiedzać pastwiska w dowolnej kolejności, zatem naturalne jest, że opłaca mu się zastosować strategię zachłanną i zaczynać od pastwisk, na których owiec jest jak najwięcej. Niech i_1, i_2, \dots, i_n będzie ciągiem numerów pastwisk uporządkowanym nierosnąco względem liczby owiec, tzn. $a[i_1] \geq a[i_2] \geq \dots \geq a[i_n]$. Zachłanny smok pożre

$$A_* = \sum_{j=1}^n \max(0, a[i_1] - j + 1)$$

owiec i będzie ucztował na pastwiskach i_1, \dots, i_k , gdzie k jest ostatnim indeksem, dla którego $a[i_k] - k + 1 > 0$. Ale, jak udowodniliśmy wcześniej, odwiedzając pastwiska ze zbioru $\{i_1, \dots, i_k\}$ w kolejności od prawej do lewej, smok pożre co najmniej A_* owiec i nie spłoszy żadnej owcy z tego zbioru. Zatem A_* jest optymalną liczbą pożartych owiec, a do jej wyznaczenia wystarczy posortować dane na wejściu ciąg, co można zrobić w czasie $O(n \log n)$.

Okazuje się, że powyższe zadanie można uogólnić, a rozwiązanie nadal będzie działać. Jeśli przedstawimy dolinę jako graf nieskierowany, w którym wierzchołki oznaczać będą pastwiska, a krawędzie łączące będą te z nich, pomiędzy którymi może przelecieć smok, to dostaniemy graf o jednej ścieżce. Możemy więc zamiast niego rozważyć dowolny graf nieskierowany. To było treścią zadania *Godzilla* z Obozu Naukowo-Treningowego im. A. Kreczmara w 2010 roku.

Tomasz IDZIASZEK