Omówienie zadań

Potyczki Algorytmiczne 2016

PARTNER Google ORGANIZATOR UNIWERSYTET WARSZAWSKI FUNDACJA ROZWOJU INFORMATYKI SPONSOR ATENDE

Turniej

Najszybsze rozwiązanie:

Mateusz Radecki (0:07)

Weźmy taki graf, że istnieje bijekcja między zawodnikami i wierzchołkami. Ponadto niech krawędź skierowana między wierzchołkami v, w nie istnieje wtedy i tylko wtedy, gdy v' na pewno przegra z w'.

Zauważmy, że gracze grają po kolei. Zawsze między wygranym i pokonanym będzie istniała skierowana krawędź.
Z tego wynika, że od zwycięzcy będzie istniała ścieżka prosta do każdego wierzchołka.

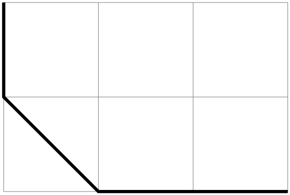
Zasymulujmy turniej. Mamy potencjalnego wygrywającego.
Z każdego innego potencjalnego wygranego musi istnieć ścieżka do wyznaczonego wygranego.
Ponadto istnienie ścieżki do wygranego z symulacji jest warunkiem dostatecznym bycia potencjalnym wygrywającym.

Łamana

Najszybsze rozwiązanie:

Kamil Dębowski (0:29)

Ile jest łamanych, których długość (po odpowiednim zaokrągleniu) jest równa l?



Liczba łamanych

Łamanych o a odcinkach pionowych/poziomych oraz b odcinkach ukośnych istnieje

$$2^a \cdot \binom{a+b}{b}$$

Liczba łamanych

Łamanych o a odcinkach pionowych/poziomych oraz b odcinkach ukośnych istnieje

$$2^a \cdot \binom{a+b}{b}$$

Wystarczy tylko wyznaczyć rozwiązania równania:

$$a + b\sqrt{2} \approx l$$

i posumować liczbę łamanych o odpowiednich parametrach.

- Gdyby założyć, że odpowiedzią nie jest `SPORO`, wtedy wystarczy sprawdzić tylko $a \in \{0, 1, 2, \ldots, 59\}$, bo dla większych a jest więcej niż 10^{18} łamanych.

- Gdyby założyć, że odpowiedzią nie jest `SPORO`, wtedy wystarczy sprawdzić tylko $a \in \{0, 1, 2, \ldots, 59\}$, bo dla większych a jest więcej niż 10^{18} łamanych.
- Dla ustalonego b zmieniając a zmienia się tylko część całkowita długości łamanej. Część ułamkowa pozostaje bez zmian. Różnych części ułamkowych po zaokrągleniu do sześciu cyfr po kropce dziesiętnej jest tylko milion.

- Gdyby założyć, że odpowiedzią nie jest `SPORO`, wtedy wystarczy sprawdzić tylko $a \in \{0, 1, 2, \ldots, 59\}$, bo dla większych a jest więcej niż 10^{18} łamanych.
- Dla ustalonego b zmieniając a zmienia się tylko część całkowita długości łamanej. Część ułamkowa pozostaje bez zmian. Różnych części ułamkowych po zaokrągleniu do sześciu cyfr po kropce dziesiętnej jest tylko milion.

Pomysł

Odwracając nieco powyższe rozumowanie: dla każdej części ułamkowej x (równej 000000, 000001, ... lub 999999) istnieje (stosunkowo nieduże) b, że $b - \lfloor b \rfloor \approx x$.

- Gdyby założyć, że odpowiedzią nie jest `SPORO`, wtedy wystarczy sprawdzić tylko $a \in \{0, 1, 2, \ldots, 59\}$, bo dla większych a jest więcej niż 10^{18} łamanych.
- Dla ustalonego b zmieniając a zmienia się tylko część całkowita długości łamanej. Część ułamkowa pozostaje bez zmian. Różnych części ułamkowych po zaokrągleniu do sześciu cyfr po kropce dziesiętnej jest tylko milion.

Pomysł

Odwracając nieco powyższe rozumowanie: dla każdej części ułamkowej x (równej 000000, 000001, ... lub 999999) istnieje (stosunkowo nieduże) b, że $b - \lfloor b \rfloor \approx x$.

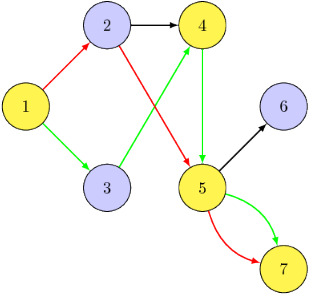
Naiwne sprawdzenie pozwala stwierdzić, że wszystkie części ułamkowe pojawią się już dla $b \approx 1.7 \cdot 10^6$. To znaczy, że dla $l \geq 2.5 \cdot 10^6$, odpowiedź jest zawsze `SPORO`.

Dwie ścieżki

Najszybsze rozwiązanie:

Mariusz Trela (1:08)

Dany jest DAG. Pytamy do jakich wierzchołków możemy dotrzeć dwoma rozłącznymi krawędziowo ścieżkami z wierzchołka o numerze 1.



Definicje

Wierzchołkiem **dobrym** nazwijmy jeden z wierzchołków o szukanej własności (dwie rozłączne krawędziowo ścieżki z 1).
Kodem ścieżki z 1 nazywamy ostatnią krawędź na niej, która wychodzi z wierzchołka dobrego.

Definicje

Wierzchołkiem **dobrym** nazwijmy jeden z wierzchołków o szukanej własności (dwie rozłączne krawędziowo ścieżki z 1).
Kodem ścieżki z 1 nazywamy ostatnią krawędź na niej, która wychodzi z wierzchołka dobrego.

Lemat

Wierzchołek v jest dobry wtedy i tylko wtedy, gdy $v = 1$, lub da się do niego dojść ścieżkami o dwóch różnych kodach.

Definicje

Wierzchołkiem **dobrym** nazwijmy jeden z wierzchołków o szukanej własności (dwie rozłączne krawędziowo ścieżki z 1).
Kodem ścieżki z 1 nazywamy ostatnią krawędź na niej, która wychodzi z wierzchołka dobrego.

Lemat

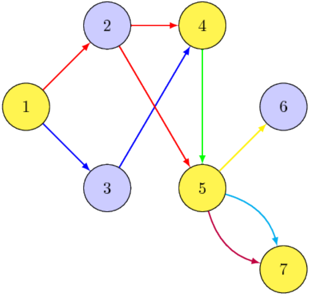
Wierzchołek v jest dobry wtedy i tylko wtedy, gdy $v = 1$, lub da się do niego dojść ścieżkami o dwóch różnych kodach.

Szkic dowodu

$\Rightarrow$: trywialne.
$\Leftarrow$: budujemy rozłączne krawędziowo (i wierzchołkowo poza dobrymi wierzchołkami) ścieżki kończące się w rozważanym dobrym wierzchołku, dobudowując kolejne fragmenty od krawędzi, których kody mamy. Trzeba rozważyć kilka prostych przypadków.

Algorytm

Rozważamy wierzchołki w porządku topologicznym. Jeżeli prowadzą do niego dwie ścieżki o różnych kodach, to na mocy lematu jest on dobry, więc zaznaczamy go takim i krawędziom od niego nadajemy nowe kody. W przeciwnym wypadku mamy ścieżkę o tym samym kodzie i przenosimy kod dalej.



Złożoność: $\mathcal{O}(n+m)$.

Osiągalność

Najszybsze rozwiązanie:

Marek Sokołowski (1:36)

Graf $G = (V, E)$ skierowany jest narysowany w kwadracie $[0, 1] \times [0, 1]$ tak, że podzbiór $Y \subseteq V$ leży na osi OY, a podzbiór jego wierzchołków X leży na osi OX.

Graf $G = (V, E)$ skierowany jest narysowany w kwadracie $[0, 1] \times [0, 1]$ tak, że podzbiór $Y \subseteq V$ leży na osi OY, a podzbiór jego wierzchołków X leży na osi OX.

Dla $y \in Y$ i $x \in X$, niech $A_{y,x} = 1$ wtedy gdy w G istnieje ścieżka $y \to x$, a $A_{y,x} = 0$ w przeciwnym przypadku.

Graf $G = (V, E)$ skierowany jest narysowany w kwadracie $[0, 1] \times [0, 1]$ tak, że podzbiór $Y \subseteq V$ leży na osi OY, a podzbiór jego wierzchołków X leży na osi OX.

Dla $y \in Y$ i $x \in X$, niech $A_{y,x} = 1$ wtedy gdy w G istnieje ścieżka $y \to x$, a $A_{y,x} = 0$ w przeciwnym przypadku.

Na ile sposobów może wyglądać macierz A?

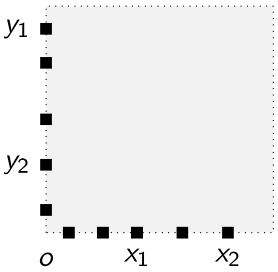
Dokładne położenia X i Y na osiach nie mają znaczenia.

Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.

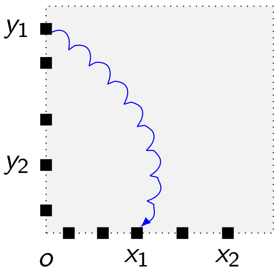
Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.



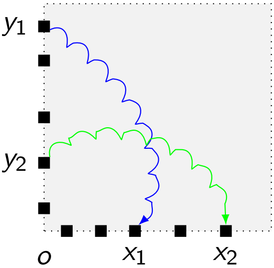
Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.



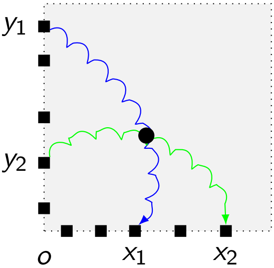
Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.



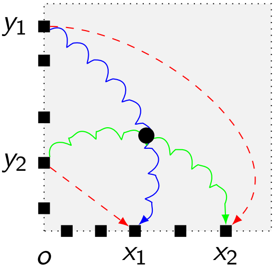
Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.



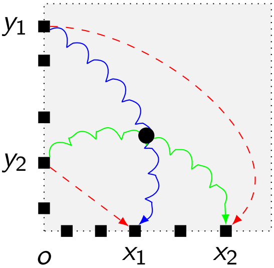
Dokładne położenia X i Y na osiach nie mają znaczenia.

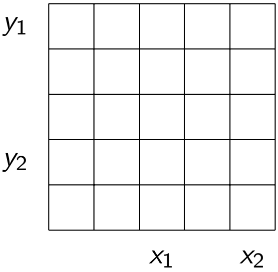
Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.



Dokładne położenia X i Y na osiach nie mają znaczenia.

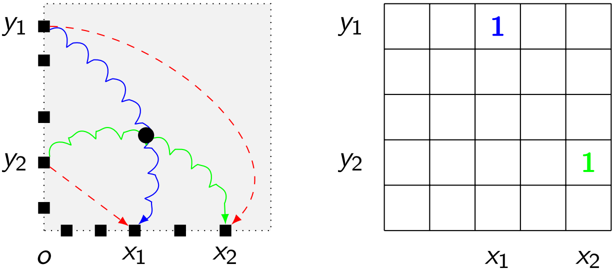
Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.





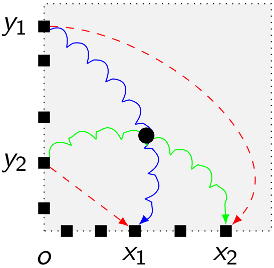
Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.



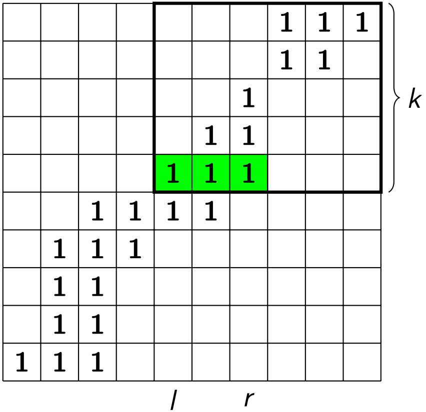
Dokładne położenia X i Y na osiach nie mają znaczenia.

Posortujmy Y w kolejności malejących współrzędnych y, a X w kolejności rosnących współrzędnych x.

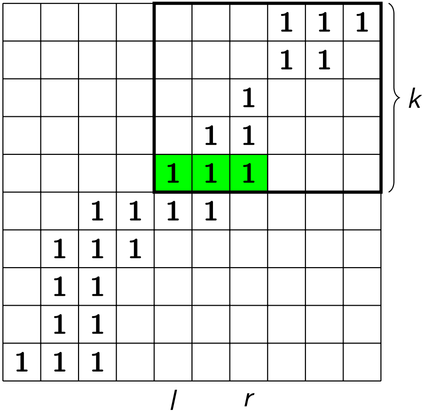


y_1			1		1
y_2			1		1
			x_1		x_2

							1	1	1
							1	1	
						1			
					1	1			
				1	1	1			
		1	1	1	1				
	1	1	1						
	1	1							
	1	1							
1	1	1							



Niech $R(k, l, r)$ oznacza liczbę macierzy rozmiaru $n \times m$ tego typu takich, że przedział jedynek w k-tym wierszu od góry są w przedziale $[l, r]$, a niższe wiersze są zgodne z $[l, r]$.



Niech $R(k, l, r)$ oznacza liczbę macierzy rozmiaru $n \times m$ tego typu takich, że przedział jedynek w k-tym wierszu od góry są w przedziale $[l, r]$, a niższe wiersze są zgodne z $[l, r]$.

Przejścia: musimy sie przeiterować po możliwych przedziałach w $(k-1)$-tym wierszu od góry.

Czyli tak naprawdę musimy zsumować wartości $R(k-1, l', r')$ takie, że $l' \geq [l, r+1]$, $r' \geq [max(l', r), m]$.

Czyli tak naprawdę musimy zsumować wartości $R(k-1, l', r')$ takie, że $l' \geq [l, r+1]$, $r' \geq [max(l', r), m]$.

Brutalne sumowanie dałoby rozwiązanie ok. $O(N^5)$, gdzie $N = max(n, m)$.

Czyli tak naprawdę musimy zsumować wartości $R(k-1, l', r')$ takie, że $l' \geq [l, r+1]$, $r' \geq [max(l', r), m]$.

Brutalne sumowanie dałoby rozwiązanie ok. $O(N^5)$, gdzie $N = max(n, m)$.

Można jednak zauważyć, że sumujemy tak naprawdę po sumie prostokątów $[l, r] \times [r, m]$ i $[r+1, r+1] \times [r+1, m]$. Dodając sumy prefiksowe, możemy obliczać przejścia w czasie stałym, dostając rozwiązanie $O(N^3)$.

Czyli tak naprawdę musimy zsumować wartości $R(k-1, l', r')$ takie, że $l' \geq [l, r+1]$, $r' \geq [max(l', r), m]$.

Brutalne sumowanie dałoby rozwiązanie ok. $O(N^5)$, gdzie $N = max(n, m)$.

Można jednak zauważyć, że sumujemy tak naprawdę po sumie prostokątów $[l, r] \times [r, m]$ i $[r+1, r+1] \times [r+1, m]$. Dodając sumy prefiksowe, możemy obliczać przejścia w czasie stałym, dostając rozwiązanie $O(N^3)$.

Puste wiersze i kolumny obsługujemy domnażając odpowiednio współczynniki dwumianowe.

Skojarzenie

Najszybsze rozwiązanie:

??? (?:??)

Dane jest drzewo. Trzeba znaleźć liczbę par wierzchołków, których połączenie zwiększy rozmiar maksymalnego skojarzenia.

Dla każdego poddrzewa i każdego poddrzewa z usuniętym korzeniem obliczmy dynamicznie:

- Wielkość maksymalnego skojarzenia.
- Liczbę wierzchołków, które mogą być niepokryte w przez takie skojarzenie.
- Liczbę par wierzchołków, które mogą być niepokryte przez takie skojarzenie.

Obliczamy również dla każdego poddrzewa:

- Liczbę wierzchołków, które mogą być niepokryte w obu tych przypadkach.
- Liczbę par wierzchołków, które mogą być niepokryte w obu tych przypadkach.

Dopasowanie

Najszybsze rozwiązanie:

??? (?:??)

Dany jest wzorzec $p_{0..m-1}$, tekst $t_{0..n-1}$ i liczba k $(0 \leq k \leq 10)$. Czy istnieje takie podsłowo w tekście, które jest prawie równe wzorcowi, czyli można przekształcić jedno w drugie poprzez ciąg co najwyżej k operacji

- wstawienia litery,
- usunięcia litery,
- zamiany litery na inną?

- Niech $D_{k,i}$ to największe s takie, że fragment wzorca $p_{0..s-1}$ możemy dopasować do fragmentu tekstu kończącego się na t_{i+s-1}, wykonując co najwyżej k operacji
- Do obliczenia wartości $D_{k,i}$ potrzebujemy wartości $D_{k-1,i-1..i+1}$ i trzech wywołań funkcji znajdującej najdłuższy wspólne podsłowo tekstu i wzorca zaczynające się od danych pozycji (np. haszowanie Karpa-Rabina albo KMR, $O(\log n)$). Złożoność czasowa $O(k(n+k)\log(n))$
- Rozwiązanie alternatywne: próbujemy zaczynać od kolejnych pól, dopasowujemy ile się da, jak się nie da dalej to próbujemy wykonać wszystkie możliwe operacje (rekurencja ze spamiętywaniem), aż zużyjemy wszystkie operacje albo dopasujemy cały wzorzec.

Waga

Najszybsze rozwiązanie:

??? (?:??)

Bajtazar posiada wagę, która wynik ważenia zaokrągla do wielokrotności c gramów lub wyświetla błąd jeśli masa wynosi co najmniej $c \cdot k$ gramów. Dany zbiór wag przedmiotów, pytanie ile par może on odróżnić wykorzystując do tego celu swoją wagę.

Lemat

Bajtazar może porównać przedmioty o wagach a i b

$$\iff$$

Istnieje podzbiór pozostałych przedmiotów o masie x, taki że wskazanie wagi w przypadku ładunku o masie $a + x$ jest inne niż dla ładunku o masie $b + x$

Lemat

Bajtazar może porównać przedmioty o wagach a i b

$$\iff$$

Istnieje podzbiór pozostałych przedmiotów o masie x, taki że wskazanie wagi w przypadku ładunku o masie $a + x$ jest inne niż dla ładunku o masie $b + x$

Dowód

$\Leftarrow$: Oczywiste

$\Rightarrow$: Jeśli zmienilibyśmy wagi przedmiotów a oraz b na $\frac{a+b}{2}$, to wynik ważenie każdego podzbioru przedmiotów pozostanie taki sam.

Lemat

Bycie nieporównywalnym jest relacją równoważności, a klasy abstrakcji tworzą spójne przedziały wag.

Lemat

Bycie nieporównywalnym jest relacją równoważności, a klasy abstrakcji tworzą spójne przedziały wag.

Dowód

Załóżmy, że a jest nieporównywalne z b, a b jest nieporównywalne z c. Po podstawieniu $\frac{a+b}{2}$ za a oraz b, jasno widać, że b jest nieporównywalne z c.

Dla danych $a \leq c \leq b$ takich, że a jest nieporównywane z b można zmienić wagę a na c, a wagę b na $a + b - c$ nie zmieniając wskazań wagi.

"Plecak bez jednego"

Obliczmy dla każdej pary sąsiednich przedmiotów i każdej reszty modulo c, jaka jest minimalna waga podzbioru pozostałych przedmiotów dająca daną resztę.

- Dodajemy dynamicznie pierwszą połowę przedmiotów i obliczmy wynik rekurencyjnie dla drugiej.
- Dodajemy dynamicznie drugą połowę przedmiotów i obliczamy wynik rekurencyjnie dla pierwszej.

Złożoność: $\mathcal{O}(nc \log n)$.

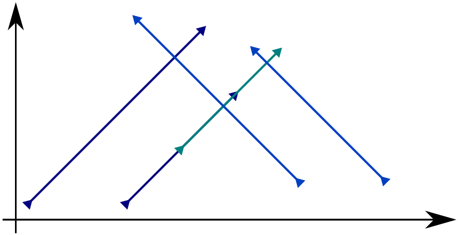
Mrówki

Najszybsze rozwiązanie:

??? (?:??)

Dane jest drzewo z wagami na krawędziach (czasy przejścia danej krawędzi przez mrówkę). Po drzewie porusza się m mrówek. i-ta mrówka pojawia się w chwili t_i w wierzchołku a_i i idzie do wierzchołka b_i. Dla każdej mrówki policz największą liczbę innych mrówek, z którymi spotka się jednocześnie podczas swojej wędrówki.

Rozważmy dla uproszczenia wersję zadania na prostej. Czyli i-ta mrówka startuje w chwili t_i z punktu a_i, i dociera do celu b_i po czasie $t_i + |b_i - a_i|$.



Po obrocie o 45° mamy odcinki poziome i pionowe. Dla każdego z nich oblicz, z iloma innymi odcinkami spotyka się on w jednym punkcie.
Wystarczy policzyć wyniki dla pionowych – dla poziomych liczy się analogicznie.

Zamiatanie od lewej do prawej – dla każdej wartości X, dla której coś się dzieje:

1. dodaj wsystkie poziome o $x_1 = X$: $t[y] + +$
2. dodaj wszystkie pionowe o $x = X$: $t[y_1..y_2] + +$
3. dla każdego odcinka pionowego oblicz wynik: $wynik = \max(t[y_1..y_2])$
4. usuń wszystkie odcinki pionowe o $x = X$: $t[y_1..y_2] - -$
5. usuń wszystkie odcinki poziome o $x_2 = X$: $t[y] - -$

Operacje na t realizujemy na drzewie przedziałowym (musimy skompresować możliwe współrzędne), czas $O(m \log m)$

Aby to uogólnić na drzewo, używamy tzw. *heavy light decomposition*.

- ukorzeniamy drzewo
- dla każdego wierzchołka drzewa łączymy krawędź do ojca z najcięższą krawędzią prowadzącą w dół, czyli tą, która prowadzi do największej liczby wierzchołków. W ten sposób rozkładamy nasze drzewo na pewną liczbę ścieżek ciężkich
- idąc od korzenia w dół możemy jedynie $O(\log n)$ razy wybrać krawędź nie będącą krawędzią najcięższą
- także każdą trasę mrówki rozkładamy na $O(\log n)$ odcinków idących ścieżkami ciężkimi, dla każdej z nich wywołujemy algorytm na prostej
- Złożoność $O(m \log m \log n)$