Akademickie Mistrzostwa Polski w Programowaniu Zespołowym

Prezentacja rozwiązań zadań

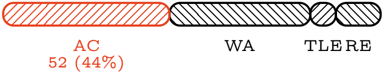
26 października 2014

Adwokat

Autor zadania: Jakub Łącki

Zgłoszenia: 118 z 857 (13%)

Zaakceptowane przez 52 z 57 drużyn (91%)



Adwokat

Mamy dane n odcinków na prostej. Należy znaleźć dwa odcinki, które się nie przecinają (lub stwierdzić, że takowych nie ma).

- Znajdujemy odcinek o minimalnym prawym końcu oraz odcinek o maksymalnym lewym końcu. Jeśli te dwa odcinki się nie przecinają, to jest to rozwiązanie.

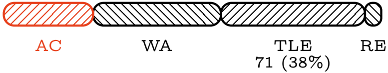
Czas działania $O(n)$.

Dzielniki

Autor zadania: Jakub Radoszewski

Zgłoszenia: 186 z 857 (21%)

Zaakceptowane przez 44 z 57 drużyn (77%)



Dzielniki

Dla danego ciągu $a_1, \ldots, a_n$ ($a_i \leq M$) chcemy znaleźć liczbę par (i, j) takich, że a_i jest dzielnikiem a_j.

- Niech $c[i]$ oznacza liczbę wystąpień i w ciągu; wyznaczamy to w czasie $O(n + M)$.
- Dla każdego $i \leq M$ rozpatrujemy wszystkie jego wielokrotności $j \leq M$ i dodajemy do wyniku $c[i] \cdot c[j]$. Wielokrotności i jest $O(M/i)$, zatem ta faza działa w czasie $O(M + \frac{M}{2} + \frac{M}{3} + \ldots + \frac{M}{M}) = O(M \log M)$.

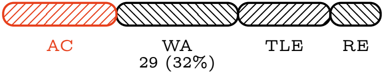
Ostateczna złożoność to $O(n + M \log M)$.

Ceny

Autor zadania: Jakub Radoszewski

Zgłoszenia: 90 z 857 (10%)

Zaakceptowane przez 27 z 57 drużyn (47%)



Ceny

Chcemy kupić $m \leq 16$ produktów, mamy do dyspozycji $n \leq 100$ hurtowni. Znamy cenę każdego produktu w każdej hurtowni oraz koszt dojazdu do hurtowni. Jaki jest najtańszy koszt wszystkich zakupów?

- Dla każdego podzbioru produktów znajdujemy najtańszy sklep, w którym możemy zakupić ten podzbiór. W tym celu rozważamy każdy podzbiór produktów w każdej hurtowni. Koszt czasowy $O(nm2^m)$ lub $O(n2^m)$.
- Dla każdego podzbioru produktów znajdujemy najtańszy koszt zakupu tych produktów. W tym celu rozważamy każdy podzbiór tego podzbioru i zakładamy, że został on kupiony w jednej hurtowni. Jest to standardowe programowanie dynamiczne po podzbiorach działające w czasie $O(3^m)$.

To daje złożoność $O(n2^m + 3^m)$.

Ceny

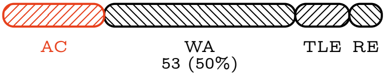
- Dla każdego podzbioru produktów S znajdujemy $c[i][S]$ – najtańszy koszt zakupu tych produktów w hurtowniach $1, \ldots, i$.
- Dodajemy hurtownię $i+1$: dla każdego podzbioru produktów S sprawdzamy, czy opłaca nam się jechać do tej hurtowni i wymienić część produktów na tańsze. Korzystamy z faktu, że dotychczasowy minimalny koszt zakupu j-tego produktu to $c[i][S] - c[i][S \setminus \{j\}]$.

Złożoność czasowa to $O(nm2^m)$.

Kapitan

Autor zadania: Jakub Łącki

Zgłoszenia: 105 z 857 (12%)
Zaakceptowane przez 28 z 57 drużyn (49%)



Kapitan

Danych jest n punktów na płaszczyźnie. Koszt przejazdu bezpośrednio między punktami (x_A, y_A) oraz (x_B, y_B) to $\min(|x_A - x_B|, |y_A - y_B|)$. Znaleźć minimalny koszt przejazdu pomiędzy dwoma zadanymi punktami.

- Jeśli mamy trzy punkty (x_A, y_A), (x_B, y_B), (x_C, y_C), $x_A \leq x_B \leq x_C$ i bezpośredni koszt przejazdu z A do C wynosi $x_C - x_A$, to koszt przejazdu nie zwiększy się, jeśli po drodze odwiedzimy punkt B.
- Z tego wynika, że z danego punktu wystarczy rozważać przejście do najbliższego punktu w każdym z czterech kierunków. Aby znaleźć punkty najbliższe wystarczy dwa razy posortować listę punktów – raz względem współrzędnej x i raz względem współrzędnej y.

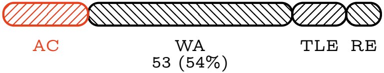
Minimalny koszt znajdujemy algorytmem Dijkstry w grafie o n wierzchołkach i co najwyżej $4n$ krawędziach w czasie $O(n \log n)$.

Euklidesowy Nim

Autor zadania: Tomasz Idziaszek

Zgłoszenia: 98 z 857 (11%)

Zaakceptowane przez 22 z 57 drużyn (38%)



Euklidesowy Nim

Na stole leży stos n kamieni. Gracze p i q grają w grę, wykonując naprzemiennie ruchy. Ruch gracza x polega na zabraniu wielokrotności kamieni x lub (jeśli nie może tego zrobić) na dołożeniu x kamieni. Wygrywa gracz opróżniający stos; wyznaczyć który.

- Jeśli n nie jest podzielne przez $d = NWD(p, q)$ to gra toczy się w nieskończoność. W przeciwnym przypadku dzielimy wszystkie liczby przez d i możemy założyć, że p i q są względnie pierwsze.
- Załóżmy, że $p < q$ (ale rozważamy jako pierwszego zarówno gracza p jak i gracza q).

Euklidesowy Nim

(A) Jeśli zaczyna q i $n < q$, to przegrywa. (Bo p za każdym razem doprowadza do sytuacji $n < p$, a q musi dołożyć q. Ponieważ $p \perp q$, to w pewnym momencie dołożenie q spowoduje, że n jest podzielne przez p.)

(B) Zatem jeśli zaczyna p i $n \geq p$, to wygrywa, bo doprowadza do sytuacji (A).

(C) Jeśli zaczyna p i $n < p$, to musi dołożyć p i albo wygrywa bo doprowadza do (A) albo $n + p \geq q$ i q musi usunąć q, co sumarycznie powoduje zmniejszenie n o $q - p$. Zatem p przegrywa wtw, gdy n jest podzielne przez $q - p$.

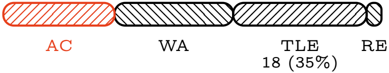
(D) Jeśli zaczyna q i $n \geq q$, to dla $z = n \bmod q$, z podzielne przez $q - p$ i $z < p$, wygrywa doprowadzając do (C). W przeciwnym przypadku przegrywa, bo dowolny ruch prowadzi do (B).

Benzyna

Autor zadania: Jakub Łącki

Zgłoszenia: 51 z 857 (5%)

Zaakceptowane przez 15 z 57 drużyn (26%)



Benzyna

Mamy nieskierowany ważony graf G o n wierzchołkach, m krawędziach i podzbiór wierzchołków (stacji). Mamy odpowiedzieć na q zapytań postaci „czy da się przejechać pomiędzy dwoma stacjami tak, by maksymalna odległość pomiędzy pośrednimi stacjami była nie większa niż δ".

- Dla każdego wierzchołka x w G znajdujemy najbliższą mu stację s_x i odległość do niej $d(s_x, x)$. Używamy do tego algorytmu Dijkstry w czasie $O(m \log n)$.
- Dla ustalonej krawędzi xy w G cysterna o baku δ może nią przejechać, gdy $w^*(xy) = d(s_x, x) + w(xy) + d(y, s_y) \leq \delta$.
- Sortujemy krawędzie względem w^* i zapytania względem δ.
- Dodajemy krawędzie do grafu, utrzymując podział na spójne składowe.

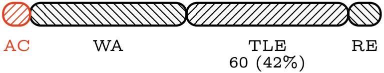
Złożoność to $O(m \log m + q \log q)$.

Globalne ocieplenie

Autorzy zadania: Jacek Tomasiewicz i Tomasz Idziaszek

Zgłoszenia: 140 z 857 (16%)

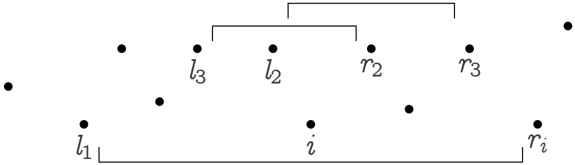
Zaakceptowane przez 10 z 57 drużyn (17%)



Globalne ocieplenie

W ciągu $a_1, \ldots, a_n$ chcemy znaleźć najdłuższy przedział zawierający jedno minimum i jedno maksimum.

- Dla każdego elementu i chcemy w czasie stałym wyznaczyć najdłuższy przedział, który zawiera ten element jako jedyne minimum. Granice tego przedziału wyznaczają l_1, r_1 będące najbliższymi elementami $\leq a_i$.



- Niech l_3, l_2, r_2, r_3 będą maksymalnymi elementami w przedziale (l_1, r_1) najbliższymi do i. Kandydatami na szukany przedział są (l_3, r_2) i (l_2, r_3).

Globalne ocieplenie

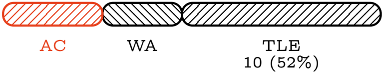
- Osobno trzeba rozważyć przypadki, gdy któryś z indeksów l_3, l_2, r_2, r_3 nie istnieje.
- Wyznaczanie indeksów można zrobić na drzewie przedziałowym w sumarycznym czasie $O(n \log n)$ lub korzystając ze stosu w czasie $O(n)$.

Jaskinia

Autor zadania: Tomasz Idziaszek

Zgłoszenia: 19 z 857 (2%)

Zaakceptowane przez 5 z 57 drużyn (8%)



Jaskinia

Dane jest drzewo o n wierzchołkach i m poddrzew: S_i składa się z wierzchołków, których suma odległości do a_i i b_i wynosi co najwyżej d_i. Należy znaleźć wierzchołek należący do przecięcia $S = \bigcap S_i$, lub stwierdzić że jest ono puste.

- Dla ustalonego wierzchołka v możemy w czasie $O(n + m)$ wyznaczyć do niego odległości i stwierdzić, czy należy on do S.
- Znajdujemy w drzewie wierzchołek v, którego wszystkie poddrzewa mają rozmiar nie większy niż $\frac{2}{3}n$. Stwierdzamy, że albo v należy do S, albo istnieje poddrzewo v zawierające S, albo S jest puste.

Wykonamy $O(\log n)$ zejść rekurencyjnych, zatem złożoność algorytmu to $O(n + m \log n)$.

Jaskinia

Można szybciej:

- Odległością v od S_i jest

$$\max(0, \lceil (d(v, a_i) + d(v, b_i) - d_i)/2 \rceil).$$

  Odległości z v do wszystkich S_i możemy wyznaczyć w czasie $O(n + m)$.
- Znajdujemy odległości z wierzchołka 1, a następnie S_i najdalsze od tego wierzchołka.
- Niech x będzie wierzchołkiem z S_i leżącym najbliżej 1. Jeśli S jest niepuste, to zawiera x.

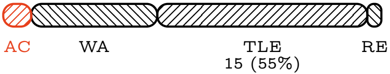
Złożoność czasowa to $O(n + m)$.

Inscenizacja

Autor zadania: Adam Karczmarz

Zgłoszenia: 27 z 857 (3%)

Zaakceptowane przez 2 z 57 drużyn (3%)



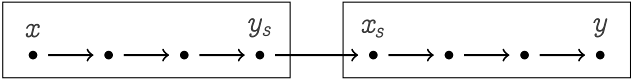
Inscenizacja

W strzelaninie uczestniczy n gangsterów; i-ty z nich odda strzał do p_i-tego. Ilu gangsterów pozostanie przy życiu dla ustalonej kolejności strzelania, oraz po każdej z q zmian w tej kolejności?

- Rozkładamy permutację $p_1, \ldots, p_n$ na cykle i każdy z nich rozpatrujemy osobno.
- Przenumerujmy gangsterów na ustalonym cyklu oraz rozwińmy ten cykl 2 razy. Jeśli i strzela pierwszy w cyklu, to $i+1$ na pewno zginie i pytamy się ilu gangsterów z przedziału $[i+1, i+n]$ przeżyje.
- Tworzymy drzewo przedziałowe, w którym operację zmiany zrealizujemy w czasie $O(\log n)$, czyli rozwiązanie będzie działać w $O((n+q)\log n)$.

Inscenizacja

- Dla przedziału gangsterów $[x, y]$ o długości 2^k pamiętamy informację ilu gangsterów z tego przedziału przeżyje oraz co się stanie z gangsterem y w zależności od tego, czy gangster x odda strzał.



- Dla przedziału $[x, y] = [x, y_s] \cup [x_s, y]$ uaktualniamy to w czasie stałym. Kluczowa jest obserwacja, że gangster x_s odda strzał jeśli $t_{x_s} < t_{y_s}$ lub gangster y_s nie przeżyje:

$$c[x, y, i] = c[x, y_s, i] + c[x_s, y, t_{x_s} < t_{y_s} \lor p[x, y_s, i] = 0]$$
$$p[x, y, i] = p[x_s, y, t_{x_s} < t_{y_s} \lor p[x, y_s, i] = 0]$$

Inscenizacja

Można prościej:

- Gangster i jest pewniakiem, jeśli $t_i < t_j$ oraz $p_j = i$.
- Jeśli odległość na cyklu do następnego pewniaka wynosi d, to przeżyje tam $\lfloor d/2 \rfloor$ gangsterów.
- Wystarczy utrzymywać zbiór pewniaków i po każdej zmianie t_i uaktualniać gangsterów i oraz p_i.

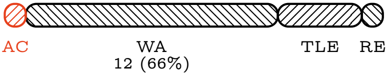
Złożoność to $O((n+q)\log n)$.

Filary

Autorzy zadania: Jakub Radoszewski i Tomasz Idziaszek

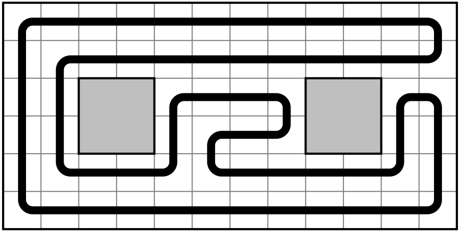
Zgłoszenia: 18 z 857 (2%)

Zaakceptowane przez 1 z 57 drużyn (1%)



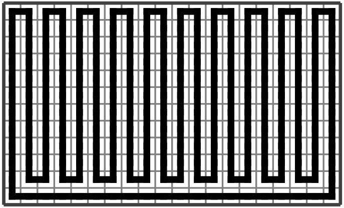
Filary

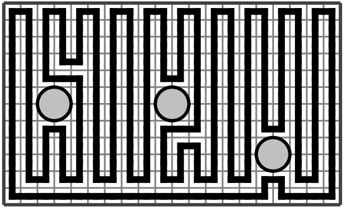
Z prostokąta o parzystych wymiarach $n \times m$ wycięto f kwadratów 2×2 tak, że środki każdych dwóch kwadratów są oddalone o co najmniej 6, a ponadto środek każdego kwadratu jest oddalony o co najmniej 3 od brzegu prostokąta. Znaleźć cykl przechodzący po wszystkich niewyciętych polach.



Filary

- Na początek wypełniamy pusty prostokąt cyklem.
- Następnie wycinamy kolejne kwadraty, lokalnie poprawiając cykl (rozważamy trzy przypadki).





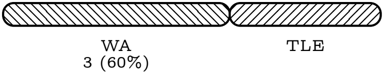
Złożoność $O(nm + f)$.

Hit sezonu

Autorzy zadania: Tomasz Kociumaka, Jakub Radoszewski, Wojciech Rytter i Tomasz Waleń

Zgłoszenia: 5 z 857 (0%)

Zaakceptowane przez 0 z 57 drużyn (0%)



Hit sezonu

Dana jest słowo długości $n \leq 3000$ nad alfabetem RGB$*$ zawierające $k \leq 19$ symboli $*$. Należy je pokryć jak najkrótszym słowem nad RGB, przy czym $*$ pasuje do wszystkiego.

- Wszystkie matryce dłuższe niż $n/2$ sprawdzamy w sumarycznym czasie $O(n^2)$.
- Rozważając krótszą matrycę będziemy zgadywać $*$ (albo w pierwszej albo w drugiej połówce słowa będzie ich co najwyżej $k/2$).
- Robimy to rekurencyjnie, utrzymując zbiór wystąpień matrycy w słowie oraz największy odstęp pomiędzy dwoma kolejnymi wystąpieniami. Jeśli odstęp nie przekracza długości matrycy, to jest ona poprawna. Uaktualnienie zbioru dla $*$ wykonujemy w czasie $O(n)$.

Czas działania to $O(n^2 + 3^{k/2}n)$.

Hit sezonu

Aby efektywnie utrzymywać zbiór wystąpień matrycy na początek wyliczamy w czasie $O(n^2)$ dla pierwszej połówki słowa P:

- Najdłuższe dopasowanie P w każdym możliwym miejscu.
- Dla każdej litery i każdej $*$ z P, wszystkie przyłożenia P, które przestają być poprawne po zastąpieniu $*$ przez tę literę.

KONIEC