Concursul de Informatică „Future for Future”
Clasa a X-a
Descrierea soluțiilor

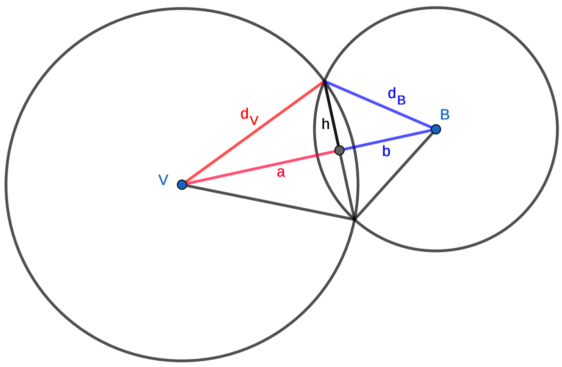
1. Problema Volei

Propusă de: Ștefan Patrichi, CNMB

Problema poate fi abordată în mai multe moduri, toate putând să obțină punctajul maxim.
În primul rând, trebuie tratate următoarele cazuri particulare:

- cercuri disjuncte exterior ($VB > d_V + d_B$) sau interior ($VB < |d_V - d_B|$)
- cercuri tangente exterior ($VB = d_V + d_B$) sau interior ($VB = |d_V - d_B|$)

Cum cercurile nu pot fi concentrice, mai rămâne cazul cercurilor secante, $|d_V - d_B| < VB < d_V + d_B$.



Soluția 1

Rezolvarea imediată (dar care se dovedește a fi și cea mai laborioasă) este cea folosind ecuația cercului. De aceea, vom prezenta doar planul general al soluției.

Vrem să aflăm soluțiile sistemului de ecuații:

$$\begin{cases} (x - x_V)^2 + (y - y_V)^2 = d_V^2 \\ (x - x_B)^2 + (y - y_B)^2 = d_B^2 \end{cases}$$

Scăzând cele două ecuații obținem una de gradul I în x și y, de tipul $Ax + By + C = 0$ (care, totodată, caracterizează dreapta pe care se află cele două puncte de intersecție). De aici, putem să scoatem o variabilă în funcție de cealaltă și să ne întoarcem în una din ecuațiile precedente. Singura observație este că trebuie să avem grijă la cazurile $A = 0$ și $B = 0$.

După ce găsim cele două perechi (x, y) care verifică, aplicăm formula distanței dintre două puncte pentru a calcula răspunsul.

Soluție de 100p: https://kilonova.ro/submissions/390370

Soluția 2

Fie P unul dintre punctele de intersecție ale cercurilor. Din teorema cosinusului în triunghiul VPB,

$$\cos V = \frac{PV^2 + VB^2 - PB^2}{2PV \cdot VB} = \frac{d_V^2 + VB^2 - d_B^2}{2d_V \cdot VB}$$

unde $VB^2 = (x_B - x_V)^2 + (y_B - y_V)^2$. Atunci,

$$2h = 2PV \sin V = 2d_V \sqrt{1 - \cos^2 V}.$$

Evident, se poate rezolva similar calculând $\cos B$.

Soluție de 100p: https://kilonova.ro/submissions/390861

Soluția 3

Aria triunghiului VPB se poate scrie:

$$\mathcal{S}[VPB] = \frac{VB \cdot h}{2} = \sqrt{p(p-VB)(p-d_V)(p-d_B)}, \text{ unde } p = \frac{VB + d_V + d_B}{2}.$$

Rezultă imediat că:

$$2h = \frac{4\sqrt{p(p-VB)(p-d_V)(p-d_B)}}{VB}$$

Soluție de 100p: https://kilonova.ro/submissions/391068

Soluția 4

Din teorema lui Pitagora,

$$\begin{cases} a^2 + h^2 = d_V^2 \\ b^2 + h^2 = d_B^2 \Rightarrow (VB-a)^2 + h^2 = d_B^2 \end{cases}$$
$$\Rightarrow a = \frac{VB^2 + d_V^2 - d_B^2}{2VB}$$

Ne întoarcem în prima ecuație și scoatem h^2. Rezultatul dorit este:

$$2h = \frac{\sqrt{4d_V^2 VB^2 - (VB^2 + d_V^2 - d_B^2)^2}}{VB}$$

Soluție de 100p: https://kilonova.ro/submissions/390197

2. Problema Polyglot

Propusă de: Ștefan Patrichi, CNMB

Notații: $\langle x, y\rangle = [x, y] \cap \mathbb{N}$, $|S| = \text{card}S$, $\binom{n}{k} = C_n^k$, v = nr. de vocale dintr-un cuvânt, c = nr. de consoane dintr-un cuvânt, $l = v + c$ = lungimea unui cuvânt, $\Sigma = 26$ = mărimea alfabetului, s = lungimea modelului unei silabe.

Observăm că ordinea consoanelor într-un model nu contează (`cCCVCc` e totuna cu `CcCVcC`). Așadar, ne interesează doar câte consoane pot exista de-o parte și de alta a unei vocale. Putem determina numerele m, n, p, q cu proprietatea că în cadrul unei silabe, orice vocală poate avea între m și n consoane în partea stângă și între p și q în partea dreaptă.

Cerința 1

Fie $a_1, a_2, \ldots, a_{v+1}$ numărul de consoane dintre perechile de vocale consecutive. Pentru ca un cuvânt să fie valid, trebuie ca $a_1 \in \langle m, n\rangle$, $a_i \in \langle m+p, n+q\rangle$ pentru $i \in \langle 2, v\rangle$ și $a_{v+1} \in \langle p, q\rangle$.

Pentru fiecare cuvânt, după ce verificăm dacă este valid, numărăm locurile dintre două consoane consecutive unde poate fi „granița” dintre silabe. Așadar, răspundem la întrebarea: în câte moduri putem scrie fiecare a_i (unde $i \in \langle 2, v\rangle$) ca sumă de două numere naturale x și y astfel încât $y \in \langle p, q\rangle$ și $x \in \langle m, n\rangle$?

Răspunsul la această întrebare este $|\langle m, n\rangle \cap \langle a_i - q, a_i - p\rangle|$. Pentru un cuvânt, înmulțim toate aceste valori corespunzătoare spațiilor intervocalice.

Complexitate timp: $O(N(l+s))$, memorie suplimentară $O(1)$.

Cerința 2

Pentru rezolvarea acestei cerințe, observăm mai întâi că vocalele și consoanele în sine nu contează, ci doar numărul lor. Trebuie să numărăm câte configurații pentru un cuvânt cu c consoane și v vocale satisfac modelul unei silabe. La final, înmulțim numărul de configurații cu numărul de permutări (cu repetiție) ale vocalelor și consonanelor.

Cum calculăm numărul de configurații? Problema ne duce cu gândul la tehnica clasică *stars and bars*, unde vocalele sunt „barele” și consoanele sunt „steluțele”. Singura diferență este că numărul de consoane pe care le putem pune în „cutii” este limitat.

Soluția 2.1

Putem folosi, pentru început, programarea dinamică. Definim `dp[i][j]` = numărul de moduri în care putem plasa j consoane în i cutii. Recurența este (`dp` are valoarea 0 pe poziții cu indecși negativi sau nemenționați mai jos):

$$\begin{cases} \texttt{dp[0][0]} = 1 \\ \texttt{dp[1][j]} = \sum\limits_{k=m}^{n} \texttt{dp[0][j - k]} \text{ pentru } j \in \langle 1, c \rangle \\ \texttt{dp[i][j]} = \sum\limits_{k=m+p}^{n+q} \texttt{dp[i - 1][j - k]} \text{ pentru } j \in \langle 1, c \rangle,\ i \in \langle 2, v \rangle \\ \texttt{dp[v + 1][j]} = \sum\limits_{k=p}^{q} \texttt{dp[v][j - k]} \text{ pentru } j \in \langle 1, c \rangle \end{cases}$$

Făcând câteva optimizări, obținem **complexitate timp** $O(Nl^2\Sigma)$ **și memorie** $O(l)$.

Soluția 2.2

Trebuie să numărăm soluțiile ecuației $a_1 + a_2 + \ldots + a_v + a_{v+1} = c$ cu constrângerile menționate mai sus. Putem mai întâi să impunem limita inferioară: adăugăm consoanele „implicite” (m pentru a_1, p pentru a_{v+1}, $m + p$ pentru restul). Așadar, am obținut ecuația echivalentă $a_1 + a_2 + \ldots + a_v + a_{v+1} = c - v(m + p) \overset{\text{not}}{=} c'$, unde limitele inferioare pentru toate variabilele sunt 0, iar cele superioare sunt: $n - m$ pentru a_1, $q - p$ pentru a_{v+1}, $n + q - m - p$ pentru restul. Fie r_i limita superioară corespunzătoare lui a_i.

Pentru a număra soluțiile, aplicăm principiul includerii și al excluderii. Ignorând limita superioară, numărul de soluții ar fi $\binom{c'+v}{v}$. În continuare, scădem soluțiile în care o variabilă depășește limita, le adunăm pe cele în care două variabile o depășesc ș.a.m.d. Obținem, astfel:

$$\sum_{S \subset \{1,2,\ldots,v+1\}} (-1)^{|S|} \binom{c' + v - \sum_{i \in S}(r_i + 1)}{v}$$

În particular, ținând cont de faptul că majoritatea limitelor superioare sunt egale, mulțimile S pot fi împărțite în:

1. $S \subset \{2, \ldots, v\}$
2. $S = \{1\} \cup$ submulțimile de la pct. 1
3. $S = \{v + 1\} \cup$ submulțimile de la pct. 1
4. $S = \{1, v + 1\} \cup$ submulțimile de la pct. 1

Răspunsul pentru submulțimile de tip 1 este:

$$\sum_{k=0}^{\left[\frac{c'}{n+q-m-p+1}\right]} (-1)^k \binom{v-1}{k} \binom{c' + v - k(n + q - m - p + 1)}{v}$$

Pentru celelalte tipuri, răspunsul se obține similar, schimbând eventual semnul (pentru tipurile 2 și 3) și adăugând la termenul superior al ultimelor combinări $-n + m - 1$ pentru tipul 2, $-q + p - 1$ pentru tipul 3, $-n - q + m + p - 2$ pentru tipul 4.

Complexitate finală timp: $O(Nl\Sigma)$, **memorie suplimentară** $O(1)$.

Soluție de 100p: https://kilonova.ro/submissions/391464