Problema 28946. Fie $f : \mathbb{R} \to \mathbb{R}$ o funcție derivabilă cu derivata bijectivă. Arătați că inversă funcție f' admite primitive.

Demonstrație. Voi demonstra că f' este continuă, deci admite primitive.

Notez P.D. = Proprietatea lui Darboux

$$\begin{cases} f' \text{ are P.D. (Teorema lui Darboux: derivata unei funcții derivabile are P.D.)} \\ f' \text{ este injectivă (bijectivitate)} \end{cases} \implies \begin{cases} f' \text{ strict monotonă} \\ f' \text{ are P.D} \end{cases}$$

$\implies f'$ continuă

$$\begin{cases} f' \text{ continuă} \\ f' \text{ inversabilă} \end{cases} \implies (f')^{-1} \text{ continuă} \implies (f')^{-1} \text{ admite primitive}$$

■

Problema S:L.24.233. Determinați mulțimea primitivelor funcției $f : \mathbb{R} \to \mathbb{R}$, $f(x) = \max(3x^2, 2x + 1)$.

Demonstrație. Pentru a determina mulțimea primitivelor funcției $f(x) = \max(3x^2, 2x+1)$, analizăm forma acesteia în funcție de valoarea lui x.

Rezolvăm ecuația pentru a determina unde funcția își schimbă ramurile:

$$3x^2 = 2x + 1$$
$$3x^2 - 2x - 1 = 0.$$

$$x = \frac{-(-2) \pm \sqrt{(-2)^2 - 4 \cdot 3 \cdot (-1)}}{2 \cdot 3}$$
$$x = \frac{2 \pm \sqrt{16}}{6} = \frac{2 \pm 4}{6}.$$

$\implies \mathrm{x}_1 = 1$ și $x_2 = -\frac{1}{3}$.

$$f(x) = \begin{cases} 3x^2 & \text{pentru } x \in \left(-\infty, -\frac{1}{3}\right] \\ 2x + 1 & \text{pentru } x \in \left(-\frac{1}{3}, 1\right] \\ 3x^2 & \text{pentru } x \in (1, \infty) \end{cases}$$

Calculăm primitivele pe fiecare interval:

- Pentru $x \in \left(-\infty, -\frac{1}{3}\right] \cup (1, \infty)$, unde $f(x) = 3x^2$:

$$F_1(x) = \int 3x^2 \, dx = x^3 + C_1.$$

- Pentru $x \in \left(-\frac{1}{3}, 1\right]$, unde $f(x) = 2x + 1$:

$$F_2(x) = \int (2x+1)\, dx = x^2 + x + C_2.$$

Mulțimea primitivelor funcției $f(x)$ este astfel dată de funcțiile $F(x)$ de forma:

$$F(x) = \begin{cases} x^3 + C_1, & x \in \left(-\infty, -\frac{1}{3}\right], \\ x^2 + x + C_2, & x \in \left(-\frac{1}{3}, 1\right], \\ x^3 + C_3, & x \in (1, \infty), \end{cases}$$

Pentru a determine o relație între constante ne putem folosi de continuitatea funcției. Pentru continuitatea funcției primitivă, impunem condițiile:

1. Pentru $x = -\frac{1}{3}$:

$$\left(-\frac{1}{3}\right)^3 + C_1 = \left(-\frac{1}{3}\right)^2 - \frac{1}{3} + C_2,$$

care se scrie ca:

$$-\frac{1}{27} + C_1 = \frac{1}{9} - \frac{1}{3} + C_2.$$

$$C_2 = \frac{5}{27} + C_1.$$

2. Pentru $x = 1$:

$$1^2 + 1 + C_2 = 1^3 + C_3,$$

adică:

$$2 + C_2 = 1 + C_3.$$

$$C_2 = -1 + C_3.$$

Rezolvând aceste ecuații, putem determina valorile constantelelor C_1, C_2, și C_3 pentru a obține o funcție primitivă continuă pe întreaga axă reală. Putem alegem $C_2 = C$ și exprima funcția primitivă astfel:

$$F(x) = \begin{cases} x^3 + C - \frac{5}{27}, & x \in \left(-\infty, -\frac{1}{3}\right], \\ x^2 + x + C, & x \in \left(-\frac{1}{3}, 1\right], \\ x^3 + C + 1, & x \in (1, \infty), \end{cases}$$

■

Problema S:L.24.226. Aflați permutările $\sigma \in S_4$ astfel încât $\sigma(\sigma(1)) = 3$ și $\sigma(\sigma(\sigma(1))) = 1$.

Demonstrație. Vom căuta permutările care verifică condițiile.
Din faptul că $\sigma(\sigma(1)) = 3$ și $\sigma(\sigma(\sigma(1))) = 1$, rezultă că $\sigma(3) = 1$.

Considerând toate permutările posibile inițial, avem $4! = 24$ cazuri. Însă, aplicând condițiile problemei, putem reduce acest număr la doar $3! = 6$ posibilități. (știind valoarea lui $\sigma(3)$)

Pentru a continua, este convenabil să enumerăm toate posibilitățile și să verificăm fiecare permutare dacă respectă condiția $\sigma(\sigma(\sigma(1))) = 1$.

Astfel, obținem următoarele permutări care îndeplinesc condițiile:

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 1 & 4 \end{pmatrix}, \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$$

Acestea sunt singurele permutări care respectă toate cerințele problemei.

Pentru a fi sigur că răspunsul este valid, am realizat un program în limbajul Python care calculează în mod automat ce permutări verifică condițiile.

```python
from itertools import permutations
n = 4

permutari = list(permutations(range(1, n + 1)))
permutari_valide = []

for permutare in permutari:
    # Deoarece Python indexează de la 0, iar permutările sunt de la 1,
    ↪ am creat o funcție lambda.
    sigma = lambda x: permutare[x - 1]
    sigma_1 = sigma(1)

    sigma_sigma_1 = sigma(sigma_1)
    sigma_sigma_sigma_1 = sigma(sigma_sigma_1)

    if sigma_sigma_1 == 3 and sigma_sigma_sigma_1 == 1:
        permutari_valide.append(permutare)

print(permutari_valide)
```

Rezultatul fiind [(2, 3, 1, 4), (4, 2, 1, 3)]. Așa cum am obținut și eu anterior.

■