

6. VNITŘNÍ REPREZENTACE VÝRAZŮ, NÁHODNÁ ČÍSLA, ZOBRAZOVÁNÍ DAT

Vnitřní reprezentace výrazu: `FullForm[výraz]` `TreeForm[výraz]`

Výraz	FullForm
$x+y$	<code>Plus[x,y]</code>
$x*y$	<code>Times[x,y]</code>
x^y	<code>Power[x,y]</code>
$\{a,b,c\}$	<code>List[a,b,c]</code>
$x \rightarrow y$	<code>Rule[x,y]</code>

Základními stavebními kameny všech výrazů jsou atomy—čísla, identifikátory a řetězce.

Příklad atomu	Head	FullForm
27	<code>Integer</code>	27
3.14	<code>Real</code>	3.14
1/2	<code>Rational</code>	<code>Rational[1,2]</code>
3+5I	<code>Complex</code>	<code>Complex[3,5]</code>
Sin	<code>Symbol</code>	Sin
"žlutý kůň"	<code>String</code>	"žlutý kůň"

Všechny výrazy, které nejsou atomy, mají tvar $f[a_1, a_2, \dots, a_n]$, kde $n \geq 0$ a f, a_1, a_2, \dots, a_n jsou buď atomy, nebo výrazy.

a_1, a_2, \dots, a_n jsou argumenty výrazu, lze je získat pomocí konstrukce `výraz[[číslo argumentu]]`.

f se nazývá hlavou výrazu, lze ji zjistit pomocí `Head[výraz]` nebo `výraz[[0]]`.

Hlavu výrazu lze změnit pomocí `Apply[nová hlava,výraz]` nebo `nová hlava@@výraz`.

Zápis $f[x]$ je ekvivalentní s postfixovým zápisem $x//f$ a s prefixovým zápisem $f@x$.

Zápis $f[x,y]$ je ekvivalentní s infixovým zápisem $x \sim f \sim y$.

Generování (pseudo)náhodných čísel:

Reálná čísla: `RandomReal[]` `RandomReal[{min,max}]` `RandomReal[{min,max},počet]`

Celá čísla: `RandomInteger[]` `RandomInteger[{min,max}]` `RandomInteger[{min,max},počet]`

Prvočísla: `RandomPrime[{min,max}]` `RandomPrime[{min,max},počet]`

Náhodný výběr ze seznamu (s opakováním): `RandomChoice[seznam,počet]`

Náhodný výběr ze seznamu (bez opakování): `RandomSample[seznam,počet]` `RandomSample[seznam]`

Nastavení semínka pseudonáhodného generátoru: `SeedRandom[n]`

Grafické zobrazování diskretních dat:

`DiscretePlot[f[i],{i,min,max}]` `DiscretePlot[f[i],{i,min,max,krok}]`

`ListPlot[{a1,a2,...}]` `ListPlot[{{x1,y1},{x2,y2},...}]` `ListPlot[{seznam1,seznam2,...}]`

Stejnou syntax mají také `ListLinePlot` a `ListLogPlot`.

`ArrayPlot[matice]` `MatrixPlot[matice]`

Vzhled obrázků lze ovlivňovat pomocí mnoha voleb (viz dokumentaci).

CVIČENÍ

1. Definujme $e = (x < 0 \ || \ x > 1) \ \&\& \ (y <= 0 \ || \ y >= 1)$. Jaká je vnitřní reprezentace výrazu e ? Vyzkoušejte `FullForm` i `TreeForm`. Co je `e[[2,1]]`?

2. Objem n -rozměrné koule o poloměru r (tj. množiny všech bodů $x \in \mathbf{R}^n$ takových, že $\|x\| \leq r$) je roven $\frac{\pi^{n/2} r^n}{\Gamma(n/2+1)}$, kde Γ je tzv. Eulerova gama funkce; v Mathematice je dostupná jako `Gamma`. Použijte `DiscretePlot` ke znázornění objemů jednotkových koulí (tj. $r = 1$) v dimenzích 1 až 30. Z obrázku vyčtěte, ve které dimenzi je objem jednotkové koule největší. Jaká je limita objemů v \mathbf{R}^n pro $n \rightarrow \infty$?

3. `Fibonacci[n]` vrací n -té Fibonacciho číslo. Zobrazte prvních 50 prvků Fibonacciho posloupnosti pomocí `ListPlot` a `ListLogPlot`. Proč vypadá druhý graf téměř jako přímka?

4. Náhodná procházka délky n je posloupnost $\{x_0, x_1, \dots, x_n\}$, kde $x_0 = 0$ a x_{i+1} vznikne z x_i přičtením náhodného reálného čísla z intervalu $[-1, 1]$. Naprogramujte funkci `randomWalk[n]`, která zkonstruuje náhodnou procházku délky n . Zobrazte pomocí `ListLinePlot` náhodnou procházku délky 100, použijte volbu `PlotMarkers->Automatic`.

Návod: Nepoužívejte rekurzi! Funkce `Accumulate` z libovolného seznamu $\{a_1, a_2, \dots, a_n\}$ vyrobí seznam částečných součtů $\{a_1, a_1 + a_2, \dots, a_1 + \dots + a_n\}$.

5. Dvourozměrná náhodná procházka délky n je posloupnost $\{\{x_0, y_0\}, \{x_1, y_1\}, \dots, \{x_n, y_n\}\}$ taková, že $\{x_0, x_1, \dots, x_n\}$ a $\{y_0, y_1, \dots, y_n\}$ jsou náhodné procházky (viz předchozí cvičení). Naprogramujte funkci `randomWalk2D[n]`, která zkonstruuje dvourozměrnou náhodnou procházku délky n . Zobrazte pomocí `ListLinePlot` náhodnou procházku délky 1000; volbou `AspectRatio->Automatic` zajistěte, aby měl obrázek stejné měřítko na obou osách.

Návod: 2D procházku je možné získat ze dvou 1D procházek pomocí `Transpose`. Také lze využít toho, že `RandomReal[{-1,1},{n,2}]` generuje n náhodných vektorů v \mathbf{R}^2 se složkami z intervalu $[-1, 1]$.

Tip: Uložíte-li náhodnou procházku (tj. seznam dvojic) do proměnné `rw`, můžete ji zobrazit ve formě animace pomocí dvojice příkazů

```
{a,b}={Min[rw],Max[rw]}; Animate[ListLinePlot[rw[[1;;i]],PlotRange->{{a,b},{a,b}},
AspectRatio->1],{i,1,Length[rw],1}].
```

6. Zvolíme-li náhodně dvojici reálných čísel $x, y \in [-1, 1]$, pak pravděpodobnost toho, že $x^2 + y^2 \leq 1$, je $\pi/4$ (dvojice x, y představuje souřadnice bodu ve čtverci $[-1, 1] \times [-1, 1]$ a pravděpodobnost, že tento bod leží uvnitř kruhu $x^2 + y^2 \leq 1$, dostaneme jako podíl obsahů kruhu a čtverce). Zkuste tuto myšlenku využít pro přibližný výpočet hodnoty π : Zvolte velké přirozené číslo n a vygenerujte n náhodných dvojic x, y . Zjistěte počet dvojic, které splňují $x^2 + y^2 \leq 1$, a vynásobte tento počet číslem $4/n$. Jak přesný odhad π dostanete, jestliže zvolíte $n = 1\,000\,000$?

Návod: Je-li `s` seznam vektorů, pak `Count[s,r-/;Norm[r]<=1]` najde počet vektorů daného seznamu, jejichž velikost je nejvýše 1.

7. `PrimePi[n]` vrací počet prvočísel menších nebo rovných n . Zobrazte pomocí `DiscretePlot` hodnoty této funkce pro $n \in \{1, \dots, 10\,000\}$; všimněte si, jak jsou prvočísla pravidelně rozložena!

K aproximaci počtu prvočísel se někdy používá Gaussův odhad $\text{PrimePi}[n] \doteq \int_2^n \frac{dt}{\log t}$, $n \geq 2$. Sestrojte pomocí `DiscretePlot` graf rozdílů $\int_2^n \frac{dt}{\log t} - \text{PrimePi}[n]$ pro $n \in \{2, \dots, 10\,000\}$.

Návod: Integrály $\int_2^n \frac{dt}{\log t}$ je potřeba počítat numericky. Použijte zabudovanou funkci `LogIntegral`, která je definována předpisem $\text{LogIntegral}[x] = \int_0^x \frac{dt}{\log t}$; Gaussův odhad pro `PrimePi[n]` tedy získáte jako `LogIntegral[n]-LogIntegral[2]`.

8. Všechna prvočísla kromě 2 jsou lichá a mají buď tvar $4k+1$, nebo $4k+3$. V obou třídách je nekonečně mnoho prvočísel. Nechť $p_1(n)$ je počet prvočísel tvaru $4k+1$ mezi prvními n prvočísly a $p_3(n)$ je počet prvočísel tvaru $4k+3$ mezi prvními n prvočísly. Platí tedy $p_1(n) + p_3(n) = n - 1$ (dvojka nepatří ani do jedné třídy) a pomocí rozdílů $d(n) = p_3(n) - p_1(n)$ můžeme sledovat, ve které třídě je více prvočísel. Použijte `ListLinePlot` k zobrazení hodnot $d(n)$ a pomocí grafu najděte nejmenší $n_0 \in \mathbf{N}$ takové, že $p_1(n_0) > p_3(n_0)$. Jaká je hodnota n_0 -tého prvočísla?

Návod: Seznam $\{d(1), d(2), \dots, d(n)\}$ lze získat pomocí `Accumulate[Mod[Prime[Range[n]],4]-2]` (rozmyslete si). Poté, co zjistíte přibližnou polohu n_0 , použijte `PlotRange->{{min,max},Automatic}` pro přiblížení části grafu mezi `min` a `max`.