Příklady složitostí

– Čas potřebný ke zpracování dat velikosti *n*, jestliže počet kroků algoritmu je dán funkcí T(n) a jestliže provedení jedné operace trvá jednu mikrosekundu.

T(n)	20	40	60	80	100
n	20μs	40μs	60μs	80μs	0.1ms
n log(n)	86μs	0.2ms	0.35ms	0.5ms	0.7ms
n^2	0.4ms	1.6ms	3.6ms	6.4ms	10ms
n^3	8ms	64ms	0.22s	0.5s	1s
n^4	0.16s	2.56s	13s	41s	100s
2^n	1s	11.7dní	36600let	3.6 10^9 let	??
$n!$	77000 let	??	??	??	??

Příklady zrychlení výpočtu

– Velikost zpracovaných dat při zachování daného času běhu programu (není uveden) a zvýšení rychlosti počítače 10x, 100x a 1000x. Původně bylo možné v tomto čase zpracovat data o velikosti 100.

T(n)	1x	10x	100x	1000x
n	100	1000	10000	100000
$n \log(n)$	100	702	5362	43150
n^2	100	316	1000	3162
n^3	100	215	464	1000
n^4	100	177	316	562
2^n	100	103	106	109
$n!$	100	100	100	101

Příklad 1

Každý ze dvou seznamů čísel je uspořádán v neklesajícím pořadí. Popište, jak vytvoříte třetí seznam, který bude obsahovat pouze taková čísla, která se vyskytují v obou daných seznamech. Musí vám stačit jeden průchod každým z daných seznamů.

Příklad 2

Seznam obsahuje kladná čísla, záporná čísla a nuly v nahodilém pořadí. Najděte takový souvislý podseznam, v němž součet všech jeho prvků je maximální možný mezi všemi souvislými podseznamy. Uvažte, zda na to může stačit jeden průchod daným seznamem.

Příklad 3

Seznam obsahuje N+1 celých čísel, každé leží v intervalu <1,N>, čísla nejsou v seznamu nijak uspořádána. Víme, že v seznamu se vyskytuje jedno číslo dvakrát, ostatní jen jednou. Určete duplikované číslo. Musí vám stačit jeden průchod seznamem a konstantně velká přidaná paměť.

Nápověda: V Pythonu to lze vyřešit jediným vhodně sestaveným příkazem, bez využití knihoven, triků apod.

Příklad 4

Je dána zafixovaná matice M o rozměru N×N. Máme postupně odpovídat na množství dotazů stejného formátu:

"Jaký je součet všech hodnot v podmatici, která má levý horní roh na pozici (i1, j1) a pravý dolní roh na pozici (i2, j2)?"

Hodnoty i1, j1, i2, j2 budou v každém dotazu obecně jiné.

K dispozici máte paměťový prostor stejně velký jako zabírá M. Určete, jaké hodnoty si musíte předpočítat, abyste na každý dotaz odpověděli v co nejkratším čase a nezávisle na velikosti hodnot i1, j1, i2, j2.

Příklad 5

V rovině je dána kružnice se středem v počátku a poloměrem R. Popište, jak určíte, kolik celočíselných mřížových bodů leží uvnitř této kružnice. Neprocházejte všechny mřížové body ve čtverecí o velikosti 2R×2R, využijte pouze mřížové body ležící blízko kružnice. Napište kompletní kód Vašeho řešení.

Příklad 6

Uvažte algoritmus násobení dvou celých čísel, tak jak je znám ze školy pro ruční násobení. Předpokládejte, že sečtení nebo vynásobení dvou *číslic* má konstantní časovou složitost.

Určete asymptotickou složitost vynásobení dvou celých čísel M, N zapsaných v desítkové soustavě.

```
Příklad násobení          9803
                        x  347
M =  9803              -------
N =  347                 68621
                        39212
                       29409
                     ---------
                       3401641
```

Příklad 7

Na obvodu kružnice jsou v libovolně nepravidelných intervalech vyznačeny body očíslované po řadě za sebou 1, 2, ..., N. Máme určit počet všech takových trojúhelníků, jejichž vrcholy leží v očíslovaných bodech a které neobsahují střed kružnice jako svůj vnitřní bod.

Navrhněte algoritmus a určete jeho asymptotickou složitost.

Řešte analogickou úlohu pro konvexní čtyřúhelníky.

Příklad 8

Na výstup máme vypsat všechna kladná celá čísla, která jsou menší než dané číslo N a která ve svém binárním zápisu obsahují právě 3 jedničky.

Jaký bude asymptotická složitost efektivního algoritmu?

Algoritmus lineární vůči N je neefektivní.

Příklad 9

Popište, jak vypočtete hodnotu

$$\log(\log(N^{N!}))$$

pro $N = 10^7$.

Jak dlouho bude trvat výpočet na Vašem osobním počítači?
Logaritmus je o základu 10.

Nepoužívejte aproximace jako např. Stirlingův vzorec apod.

Příklad 10

Symbolem lg značíme logaritmus o základu 2.

Uspořádejte podle řádu růstu uvedené funkce poměnné n.

Zdůvodněte pořadí každých dvou sousedních funkcí v tomto uspořádání.

$$\lg(n!) \qquad \left(\sqrt{2}\right)^{\lg(n)} \qquad 2^{\lg(\lg(n))} \qquad 4^{\lg(n)} \qquad \sqrt{\lg(n)} \qquad n \cdot \lg(n^2) \qquad n \cdot \lg(n) \qquad \lg(n)^2$$