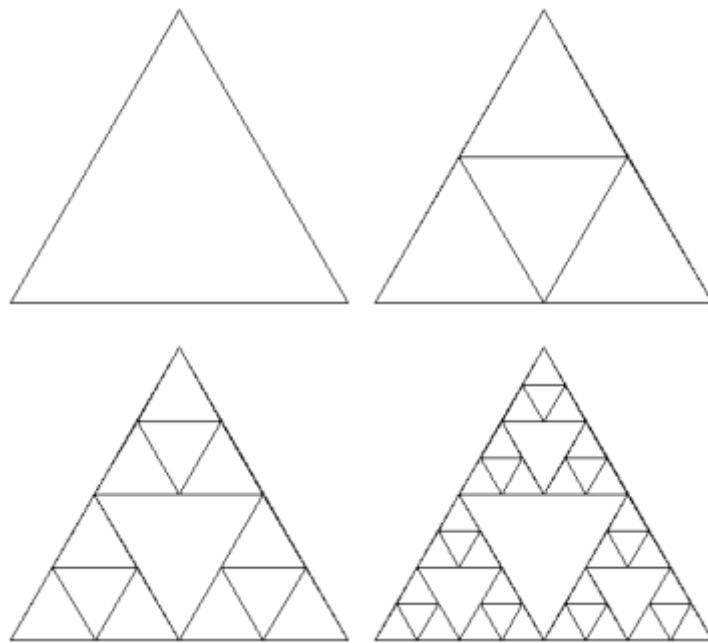


3. Algoritmizace - Rekurze, Brute Force, Heuristiky, Nedeterministky

Rekurze

v oblasti matematiky rekurzi chápeme jako definování objektu pomocí sebe sama.

Např. Sierpinského trojúhelník - je tvořen jen pomocí trojúhelníků



Rekurze v programování

Metoda volá samu sebe.

Hlavní věc v rekurzivní metodě je příkaz na zavolání samu sebe.

Metoda, která volá sama na sebe je "rekurzivní".

Všechny případy rekurzí lze převést na nerekurzivní algoritmus

Rekurze musí obsahovat podmínku ukončení (konečnost algoritmu), jinak by byla nekonečná. To může být způsob zefektivnění rychlosti nějakých programů. Rekurze je spojená s vysokými nároky na využití paměti zásobníku.

Příklady

- strom (z větve jdou větve)
- většina her - pravidla v každém tahu jsou stejná, princip řešení je tedy stejný i když se data mění.

Využití

Rekurze se dá jednoduše použít při počítání faktoriálu.
Rekurzivní chování může být různé v závislosti na tom, kolik podprogramů se jí účastní.
Metoda f1 je volána zatímco jedno z předchozích volání f1 ještě nebylo ukončeno návratem. Vzniká nám v paměti díky tomuto tzv. rozpracovanost metody a při každé další iteraci se nám ukládají proměnné průchodu, čímž vzniká velká náročnost na paměť.

Přímá rekurze

Přímá rekurze nastává, když podprogram volá přímo sám sebe. = volá se přímo

Nepřímá rekurze

Nepřímá rekurze je situace, kdy vzájemné volání podprogramů vytvoří „kruh“. Např. ve funkci A se volá funkce B a ve funkci B se volá opět funkce A. = volá se zprostředkovaně přes jinou funkci

Podprogram může být volán jednou nebo vícekrát:

1. Lineární rekurze nastává, pokud podprogram při vykonávání svého úkolu volá sama sebe pouze jednou. Vytváří se takto lineární struktura postupně volaných podprogramů.

Třeba faktoriál, nebo hledání hodnoty v setříděném poli

1. Stromová rekurze nastává, pokud se funkce nebo procedura v rámci jednoho vykonání svého úkolu vyvolá vícekrát. Strukturu volání je možné znázornit jako zakořeněný strom. Pro dvě volání v jednom průchodu vzniká binární strom, pro tři ternární strom, atd. (Počet rekurzivních volání nemusí být konstantní, např. při rekurzivním procházení grafu voláme zpracování na všechny sousedy vrchołu, a těch je obecně různý počet.)

Třeba třídící algoritmy

Druhy

Příklady

Brute Force

Heurestiky

Deterministické algoritmy

Příklad

Nedeterministické algoritmy

Příklad

