

3. Algoritmizace - Rekurze, Brute Force, Heuristiky, Nedeterministické algoritmy

Rekurze

Rekurze je případ, kdy je objekt součástí sebe samotného. Můžeme si to představit jako funkci, která se provede a následně na konci zavolá samu sebe. Tím pádem se uzavře do nekonečného cyklu. Je vhodné někde ve funkci kontrolovat, jestli je nějaká podmínka splněna, v tom případě by nastal return a konec rekurzivní metody.

```
public static void rekurze(int i) {  
    if (i < 5) {  
        System.out.println("Hodnota i je: " + i);  
        rekurze(i + 1);  
    }  
}
```

Rekurze může být výhodná pro prohledávání datové struktury strom, případně třeba k hledání nejkratší cesty v grafu.

Jednoduchým příkladem využití rekurze je právě prohledávání stromu, ve kterém zavoláme rekurzivní metodu na hlavní kořen, který pro každou „Child Node“ – odnož zavolá tu samou metodu. Tím docílíme toho, že projdeme každý prvek.

Brute Force

Brute force je technika procházení, která spočívá ve využití hrubé síly – tím pádem jsou postupně procházeny všechny prvky bez jakékoliv logiky. Jedná se o striktně deterministický algoritmus.

Pokud je ovšem počet dat malý, je například výhodnější než Merge Sort – třeba při sortování obyčejného pole. Při procházení například stromem jsou ovšem lepší sorty, které přestanou řešit nějakou část stromu, pokud nebude vyhovovat, jelikož brute force by projel i ty.

Je extrémně náročný, jak časově, tak paměťově, na druhou stranu je ale většinou velmi primitivní a vždy najde řešení (pokud je naprogramován správně)

Využívá se například na prolomení hesel typu PIN, kde je 10000 možností a víme, že jedna z nich je 100% správně, prolomení hashování a nebo již zmíněné sortování.

Heuristika

Heuristika je řešení problému, na který neznáme algoritmus, pomocí logiky. Často je takovéto řešení letmé, založené na zkušenostech a nebo intuici. Jejím hlavním cílem je najít co nejpřesnější výsledek v co nejkratším čase pro konkrétní data a proto je možné, že při jiných datech by algoritmus nefungoval.

Příkladem může být řešení rozesazení lidí do lodiček podle váhy. **Bruteforce** by vygeneroval stavový prostor a řešil by, kolik váží levá a pravá strana, načež by vybral nejvhodnější možnost. **Monte Carlo**, nedeterministický algoritmus, by několikrát náhodně rozsadil a poté vybran nejvhodnější možnost. **Heuristický** algoritmus by nejdříve všechny podle váhy seřadil a následně je řadil postupně doleva, doprava, doleva, doprava.

Deterministické a nedeterministické algoritmy

Deterministický algoritmus je takový algoritmus, který za stejných podmínek vrátí stejný výsledek.

Nedeterministický je jeho opak, ten stejnou hodnotu nevrátí. Může mít totiž krok, který má více možností, zatímco deterministické je pevně stanovené mají. Příkladem může být například algoritmus Monte Carlo, které pracuje jenom s přibližným výsledkem, jelikož má pokaždé jiná data na vstupu.

Monte Carlo

Monte Carlo je algoritmus, který pracuje jenom s určitou částí dat. Pokud máme třeba 100 hodnot a máme vybrat nejmenší, Monte Carlo náhodně zvolí například 30 hodnot a z nich vrátí přibližný výsledek. Proto může být nejmenší číslo 1 a v dalším případě 50.