

Bubble sort

Jednoduchý algoritmus, který se používá hlavně pro výukové účely a nenáročné aplikace.

Princip

Algoritmus opakovaně prochází seznam, přičemž porovnává každé dva sousedící prvky, a pokud nejsou ve správném pořadí, prohodí je.

Časová složitost

Tento algoritmus řazení je jedním z nejpomalejších, je horší i ve srovnání s jinými algoritmy se stejnou asymptotickou složitostí, neboť vyžaduje velké množství zápisů do paměti a tím neefektivně pracuje s cache procesoru.

Výhody

- Nejjednodušší algoritmus pro řazení
- Je stabilní, nemění pozice prvků, které jsou při porovnávání vyhodnoceny jako ekvivalentní
- Stačí mu sekvenční vstup k datům

Nevýhody

- Nevhodné pro řazení velkých polí prvků
- Zdlouhavé

Quick sort

Jeden z nejrychlejších běžných algoritmů založených na porovnávání prvků

Princip

Algoritmus rozděluje pole na dvě přibližně stejné části. V jedné části jsou čísla větší a ve druhé menší. Poté je zvolena hodnota „pivot“, která kontroluje, jestli jsou obě části přibližně stejné. Poté se obě části seřadí bude seřazené také pole. Obě části se pak rekurzivně řadí stejným způsobem.

Výhody

- Mnohem rychlejší než ostatní řadící algoritmy
- Při správné implementaci prakticky nepotřebuje dodatečnou paměť, řadí prvky přímo v poli.
- Bývá implementován v některých systémových knihovnách

Nevýhody

- Jde o nestabilní algoritmus, špatná volba bodu pivot může mít vliv na řazení
- Nemusí být vhodný pro slabší hardware

Selection sort

Algoritmus, který je většinou používán pro uspořádání malých množství dat

Princip

Pole se rozdělí na seřazenou a neseřazenou část. Najde se nejmenší prvek v neseřazené části a nahradí se na první pozici seřazené části. Nejmenší prvek je poté z neseřazené části odebrán. Zbytek algoritmu poté pokračuje stejným principem.

Výhody

- Je univerzální
- Pracuje lokálně, nepotřebuje pomocnou paměť
- Vhodný pro malé množství dat

Nevýhody

- Je nestabilní
- Nepatří mezi přirozeně řadící algoritmy

Insertion sort

Řadící algoritmus založený na porovnání

Princip

Algoritmus prochází prvky postupně a každý další neseříděný prvek zařadí na správné místo do již seřazené posloupnosti.

Výhody

- Jednoduchá implementace
- Efektivní na malých množinách
- Řadí stabilně

Nevýhody

- Je pomalejší než pokročilé algoritmy
- Nevhodný pro velké pole dat

Heap sort

Jeden z nejlepších obecných algoritmů řazení, založený na porovnání prvků

Princip

Využívá datové struktury **halda**. Halda umí efektivně provést operaci vložení prvku a operaci výběr největšího prvku. Proto lze pomocí haldy seřadit dodaná data od největšího k nejmenšímu, prostě pomocí vložení a následného postupného vybírání největšího prvku.

Výhody

- Stavěný pro velké množství dat
- Nepotřebuje paměť navíc
- Stabilní algoritmus

Nevýhody

- Není vhodný pro menší počet prvků

Bogo sort

Algoritmus, který je doslova vtip.

Princip

Algoritmus zkouší náhodné kombinace prvků v poli, než se strefí do správné kombinace.

Výhody

- Žádné nejsou

Nevýhody

- Strašně dlouhá časová náročnost
- Jelikož je založen na náhodném tipování, tak některé tipy, které už zkusil může provést znovu
- Absolutně zbytečný

Stalin sort

Velice efektivní algoritmus, který využívá eliminaci.

Princip

Jde v poli postupně, pokud je další prvek v pořadí, tak ho eliminuje (smaže).

Výhoda – Rychlost

Nevýhoda – Možná ztratíte některá data