

Quick sort

Maschita Jan

Co je to Quick sort?

- Algoritmus pro řazení dat založený na principu "rozděl a panuj"
- Jeden z nejrychlejších řadících algoritmů v praxi.
- Užití:

Databáze: Pro rychlé třídění záznamů.

Grafika: Pro třídění objektů při vykreslování.

Systémy souborů: Rychlé řazení seznamů souborů.

Základní myšlenka

- Quick Sort rozdělí seznam na menší části, které se postupně třídí. Dělí seznam na základě takzvaného **pivota** a pracuje rekurzivně, dokud není seznam kompletně seřazený.

Princip Quick sortu

1. Pivot (číslo rozdělující data):

Pivot je prvek seznamu, kolem kterého budou ostatní prvky uspořádány.

Obvyklé metody výběru pivotu:

- První prvek seznamu

- Poslední prvek seznamu

- Střední prvek seznamu

- Náhodný prvek

Princip Quick sortu

2. Rozdělení seznamu

Prvky menší než pivot jsou přesunuty na levou stranu.

Prvky větší než pivot jsou přesunuty na pravou stranu.

Pivot je na své správné finální pozici.

Princip Quick sortu

3. Rekurzivní třídění

Quick sort se volá rekurzivně na:

Levou část

Pravou část

- Tento proces pokračuje, dokud nezbývají části s jedním nebo žádným prvkem (ty už jsou automaticky seřazeny).

Příklad na seznamu

- Příklad seznamu: **[8, 3, 0, 7, 1, 10, 2]**
- **První iterace:**
- Pivot: 7 (poslední prvek).
- Rozdělení:
 - Menší než 7: [3, 0, 1, 2]
 - Pivot: [7]
 - Větší než 7: [8, 10]
- Výsledek: **[3, 0, 1, 2], [7], [8, 10]**

Příklad na seznamu

- **Na levé části** [3, 0, 1, 2]
se zvolí nový pivot, např. 2:
 - Menší než 2: [0, 1]
 - Pivot: [2]
 - Větší než 2: [3]
- **Na pravé části** [8, 10]
se zvolí pivot, např. 10:
 - Menší než 10: [8]
 - Pivot: [10]
 - Větší než 10: []
- **Spojení výsledků:**
 - Kompletně seřazený seznam: **[0, 1, 2, 3, 7, 8, 10]**

Časová náročnost

- **Průměrný případ:** $O(n \log n)$ – Dobrý výběr pivotu.
- **Nejhorší případ:** $O(n^2)$ – Špatný výběr pivotu (např. pivot je vždy nejmenší nebo největší prvek).

Výhody Quick sortu

- **Rychlost v praxi:**

Quick Sort je často rychlejší než jiné algoritmy

- **Široká použitelnost:**

Vhodný pro většinu praktických problémů

- **Rozdělení seznamu na místě:**

Nevyžaduje rozdělení seznamu do samostatných polí, což šetří paměť a zrychluje třídění.

Nevýhody Quick sortu

- **Nestabilní třídění:**

Při stejných hodnotách může dojít ke změně jejich relativního pořadí, což může být problém u datových struktur s více atributy.

- **Citlivost na výběr pivotu:**

Špatná strategie výběru pivotu může výrazně zpomalit algoritmus.

- **Nevhodný pro malé seznamy:**

Pro malé seznamy je často méně efektivní než jednoduché algoritmy jako Bubble Sort nebo Insertion Sort.

Děkuji za pozornost ❤️



- **Video:** [Quick Sort \(LR pointers\) - YouTube](#)