

Evolve QuickSortu: Původní algoritmus

Sort(X) :

0. Pokud $|X| \leq 1$, vrátíme X .
1. Vybereme prostřední prvek množiny X jako pivota p .
2. $M \leftarrow \{x \in X : x < p\}$,
 $P \leftarrow \{x \in X : x = p\}$,
 $V \leftarrow \{x \in X : x > p\}$.
3. $M \leftarrow \textit{Sort}(M)$,
 $V \leftarrow \textit{Sort}(V)$.
4. Vrátíme $M + P + V$.

Čas: $\mathcal{O}(n \log n)$ průměrně

Paměť: $\mathcal{O}(n)$ na pomocná pole, $\mathcal{O}(n)$ na zásobník

Evoluce QuickSortu: Třídění na místě

$Sort(X, a, b)$: (setřídí prvky $X[a], \dots, X[b]$)

0. Pokud $a \geq b$, vrátíme se.
1. $m \leftarrow \lfloor (a + b)/2 \rfloor$, $p \leftarrow X[m]$. (vybereme pivota)
2. Přeházíme prvky tak, aby nalevo byly $\leq p$, napravo $\geq p$:
3. $l \leftarrow a$, $r \leftarrow b$.
4. Dokud $l \leq r$, opakujeme:
 5. Dokud $X[l] < p$: $l \leftarrow l + 1$.
 6. Dokud $X[r] > p$: $r \leftarrow r - 1$.
 7. $X[l] \leftrightarrow X[r]$.
 8. $l \leftarrow l + 1$, $r \leftarrow r - 1$.
9. $Sort(X, a, r)$, $Sort(X, l, b)$.

Evoluce QuickSortu: Zbavíme se rekurze

$Sort(X)$:

0. Pokud $n = |X| \leq 1$, skončíme rovnou.
1. $S \leftarrow \{(1, n)\}$. (náš vlastní zásobník)
2. Dokud $S \neq \emptyset$, opakujeme:
3. Vybereme (a, b) z S .
4. $m \leftarrow \lfloor (a + b)/2 \rfloor$, $p \leftarrow X[m]$.
5. Přeházíme prvky $\dots \rightarrow l, r$.
6. Pokud $a < r$, přidáme (a, r) do S .
7. Pokud $l < b$, přidáme (l, b) do S .

Evoluce QuickSortu: Omezíme spotřebu paměti

$Sort(X)$:

0. Pokud $n = |X| \leq 1$, skončíme rovnou.
1. $a \leftarrow 1, b \leftarrow n, S \leftarrow \emptyset$.
2. Opakujeme: (právě třídíme $X[a], \dots, X[b]$, S je zásobník)
3. Vybereme pivota a přeházíme prvky $\dots \rightarrow l, r$.
4. Pokud $r - a > b - l$, prohodíme $(a, r) \leftrightarrow (l, b)$.
(interval (a, r) je teď ten menší)
5. Pokud $r > a$: (oba intervaly jsou netriviální)
6. Přidáme (l, b) do S . (větší na zásobník)
7. $(a, b) \leftarrow (a, r)$. (pokračujeme menším)
8. Jinak pokud $b > l$: $(a, b) \leftarrow (l, b)$. (větší netriviální)
9. Jinak: (oba triviální)
10. Pokud $S = \emptyset$, skončíme.
11. Jinak odebereme (a, b) z S .

Nyní stačí $\mathcal{O}(\log n)$ paměti pro zásobník.

Evoluce QuickSortu: Zkřížíme s InsertSortem

$Sort(X, K)$: (K je vhodná konstanta)

0. Pokud $n = |X| \leq K$, setřídíme InsertSortem.
1. $a = 1, b = n, S = \emptyset$.
2. Opakujeme:
3. Vybereme pivota a přeházíme prvky $\dots \rightarrow l, r$.
4. Pokud $r - a > b - l$, prohodíme $(a, r) \leftrightarrow (l, b)$.
5. Pokud $r - a > K$: (oba intervaly jsou netriviální)
6. Přidáme (l, b) do S . (větší na zásobník)
7. $(a, b) \leftarrow (a, r)$. (pokračujeme menším)
8. Jinak pokud $b - l > K$: $(a, b) \leftarrow (l, b)$. (větší netriviální)
9. Jinak: (oba triviální)
10. Pokud $S = \emptyset$, skončíme.
11. Jinak odebereme (a, b) z S .
12. Dotřídíme posloupnost InsertSortem.
(každý prvek se posune o nejvýše K , takže je to lineární)