

Základné algoritmy

OBSAH

1. Riešenie numerických úloh
 - hľadanie minima, maxima, priemeru
 - práca s maticami
2. Triedenie
 - priamym vkladáním
 - priamym výberom
 - priamou výmenou (bublinové triedenie)
3. Vyhľadávanie
 - binárne vyhľadávanie

Základné algoritmy – riešenie numerických úloh

- Hľadanie minima
- Hľadanie maxima
- Výpočet priemeru
- Práca s maticami

Triedenie

- **Def.:** Proces preusporiadania danej množiny objektov v špecifickom poradí.
- Účelom triedenia je uľahčiť neskoršie vyhľadávanie prvkov triedenej množiny.
- Dôležitá a základná činnosť v spracovaní údajov (objekty sa triedia v telefónnych zoznamoch, registroch daní z príjmu, obsahoch kníh, knižniciach, slovníkoch,...)
- Veľká závislosť výberu algoritmu od štruktúry údajov \Rightarrow obvyklé základné **delenie** na:
 - **Triedenie polí** (*vnútorné triedenie* – polia sa uchovávajú vo vnútorných pamätiach počítačov)
 - **Triedenie** (sekvenčných) **súborov** (*vonkajšie triedenie* – súbory sú umiestnené na väčších a pomalších vonkajších pamäťových médiách – disky/pásy)

Triedenie – pojmy a zápis

- Máme prvky: a_1, a_2, \dots, a_n
- Triedenie s počíva v permutovaní týchto prvkov na také poradie: $a_{k1}, a_{k2}, \dots, a_{kn}$,
že ak je daná funkcia usporiadania f , platí:
$$f(a_{k1}) \leq f(a_{k2}) \leq \dots \leq f(a_{kn})$$
- Zvyčajne sa funkcia usporiadania nevyhodnocuje podľa špecifikovaného pravidla výpočtu, ale je uložená ako explicitná zložka (časť) každého prvku = **klúč**
- Na zobrazenie prvkov a_i sa teda mimoriadne hodí štruktúra záznamu pozostávajúca z kľúča (napr. typu celé číslo s predpokladaným úmyslom identifikovať prvky) a zvyšných zložiek (pre účely triedenia nepodstatné)
- Meradlo efektívnosti metód triedenia: počet porovnaní kľúčov (C) + počet presunov prvkov (M)

Triedenie polí - priame metódy

Dôvody použitia:

1. Vhodné na objasnenie charakteristických črt hlavných zásad triedenia.
2. Ich programy sú ľahko pochopiteľné a krátke.
3. Rýchlejšie pre dostatočne malé n , nepoužívajú sa však pre veľké n .

Rozdelenie na kategórie:

1. Triedenie vkladáním.
2. Triedenie výberom.
3. Triedenie výmenou.

Triedenie priamym vkladáním

- Prvky sa rozdelia na dve skupiny: zdrojovú (a_i, \dots, a_n) a cieľovú (a_1, \dots, a_{i-1})
- V rámci každého kroku algoritmu sa zo zdrojovej postupnosti vyberie i -ty prvok a vloží sa na patričné miesto cieľovej postupnosti

- vid' vývojový diagram
- $C_{\min} = n - 1$; $C_{\max} = \sum_{i=2}^n (i-1) = \sum_{i=1}^{n-1} i = n * (n-1) / 2 = (n^2 - n) / 2$

$$C_{\text{priem}} = \sum_{i=2}^n i / 2 = (n(n+1) / 2 - 1) / 2 = (n^2 + n - 2) / 4$$

- $M_{\min} = 2(n - 1)$; $M_{\max} = \sum_{i=2}^n i = (n(n+1) / 2 - 1) = (n^2 + n - 2) / 2$

$$M_{\text{priem}} = \sum_{i=2}^n (i / 2 + 1) = n(n+1) / 4 - 1 / 2 + n - 1 = (n^2 + 5n - 6) / 4$$

Triedenie priamym výberom

1. Vyber prvok s najmenším kľúčom.
2. Vymeň ho s prvým prvkom a_1 .
3. Opakuj postupne s $n-1$ prvkami, $n-2$ prvkami,... Až ostane jediný najväčší prvok.

- vid' vývojový diagram

- $C = (n^2 - n) / 2$

$$M_{\min} = 3(n-1); \quad M_{\max} = \text{trunc}\left(\frac{n^2}{4}\right) + 3(n-1)$$

$$M_{\text{priem}} = n(\ln n + \gamma) \quad \text{Eulerova konšt. } \gamma = 0,577216\dots$$

- Vo všeobecnosti je toto triedenie efektívnejšie ako triedenie priamym vkladáním (až na postupnosti usporiadané resp. takmer usporiadané)

Triedenie priamou výmenou

- tzv. bublinové triedenie – postupné porovnávanie a zámena susedných dvojíc, až kým nie je pole utriedené
- vid' vývojový diagram
- $C = (n^2 - n) / 2$
 $M_{\min} = 0; \quad M_{\max} = \frac{3}{2}(n^2 - n)$
 $M_{\text{priem}} = \frac{3}{4}(n^2 - n)$
- Vo všeobecnosti je toto triedenie najmenej efektívne z uvádzaných algoritmov triedenia

Iné algoritmy vnútorného triedenia

- Triedenie vkladáním so zmenšovaním kroku (Shellovo triedenie)
- Stromové triedenie (bude uvedené neskôr pri dynamických údajových štruktúrach)
- Vylepšené triedenie výmenou – rýchle triedenie (quicksort) – najlepšia metóda triedenia v poli (bude uvedená neskôr pri preberaní rekurzie)

Vyhľadávanie

- Nájdenie prvku poľa (napr. min, max, zadaného prvku) – v neusporiadanom poli – vid' príklady z úvodu prednášky
- Vyhľadávanie sa dá podstatne urýchliť, ak sú prvky poľa usporiadané \Rightarrow najzaužívanejšou technikou je opakované delenie intervalu na podintervaly, v ktorých sa hľadaný prvok hľadá \Rightarrow *binárne vyhľadávanie*
- vid' vývojový diagram