

Algoritmy a Dátové Štruktúry

Jana Katreniaková

`katreniakova@dcs.fmph.uniba.sk`

Abstraktná dátová štruktúra slovník

- Postupnosť objektov - k hodnote priradíme kľúč
- Kľúč dopočítavame!
- Operácie hľadanie prvku (podľa kľúča), vloženie prvku (dvojice), vymazanie prvku (podľa kľúča)...

Možnosti implementácie

- Kompaktné množiny - polia
- Spájané zoznamy
- Hashovacie tabuľky

Abstraktná dátová štruktúra slovník

- Postupnosť objektov - k hodnote priradíme kľúč
- Kľúč dopočítavame!
- Operácie hľadanie prvku (podľa kľúča), vloženie prvku (dvojice), vymazanie prvku (podľa kľúča)...

Možnosti implementácie

- Kompaktné množiny - polia
- Spájané zoznamy
- Hashovacie tabuľky

Množiny – kompaktné pole

- Ukladáme dvojice (hodnota, kľúč)
- Vkladanie: na koniec $O(1)$
- Vyberanie: potrebujeme nájsť zodpovedajúci kľúč $O(N)$

Utriedené pole

- Utriedené podľa hodnoty kľúča
- Vkladanie, vyberanie: $O(N)$ (nemôžeme nechať medzery a udržať poradie)
- Vyhľadávanie: $O(\log N)$

Množiny – kompaktné pole

- Ukladáme dvojice (hodnota, kľúč)
- Vkladanie: na koniec $O(1)$
- Vyberanie: potrebujeme nájsť zodpovedajúci kľúč $O(N)$

Utriedené pole

- Utriedené podľa hodnoty kľúča
- Vkladanie, vyberanie: $O(N)$ (nemôžeme nechať medzery a udržať poradie)
- Vyhľadávanie: $O(\log N)$

Spájané zoznamy

- HighestPriority: $O(\log N)$ - vieme kde je minimum resp. maximum ale kým sa k nemu dostaneme...
- Vkladanie: $O(\log N)$
- Vyberanie HighestPriority: $O(\log N)$ - ako teda každého iného prvku

Hashovacie tabuľky

Nech K je množina všetkých kľúčov a tabuľka je veľkosti N .

Hashovacia funkcia

- Transformuje kľúče na indexy podľa $h : K \rightarrow \{0 \dots N - 1\}$. Teda povie, kam prvok s daným kľúčom patrí v tabuľke.
- Jednoduchá ale pritom by nemala pridelať často rovnaké indexy (rovnomerná distribúcia kľúčov do tabuľky).
- Často sa používa funkcia $h(k) = k \bmod N$ (je dobré, ak v tomto prípade N je prvočíslo nie blízko mocniny 2)

Jednoduché riešenie

- Vyhľadávanie (podľa kľúča): Hash kľúča nám povie miesto v tabuľke, kde by prvok mal byť. A buď tam je alebo nie.
- Vymazanie prvku s daným kľúčom: Ak taký prvok v tabuľke je, bude na mieste $h(K)$. Ak je to on, vymažeme ho.

Hashovacie tabuľky

Nech K je množina všetkých kľúčov a tabuľka je veľkosti N .

Hashovacia funkcia

- Transformuje kľúče na indexy podľa $h : K \rightarrow \{0 \dots N - 1\}$. Teda povie, kam prvok s daným kľúčom patrí v tabuľke.
- Jednoduchá ale pritom by nemala prideliť často rovnaké indexy (rovnomerná distribúcia kľúčov do tabuľky).
- Často sa používa funkcia $h(k) = k \bmod N$ (je dobré, ak v tomto prípade N je prvočíslo nie blízko mocniny 2)

Jednoduché riešenie

- Vyhľadávanie (podľa kľúča): Hash kľúča nám povie miesto v tabuľke, kde by prvok mal byť. A buď tam je alebo nie.
- Vymazanie prvku s daným kľúčom: Ak taký prvok v tabuľke je, bude na mieste $h(K)$. Ak je to on, vymažeme ho.

TBD