

Anotace

- Předání funkce v parametru,
- opakování (vnější třídění, medián lineárně),
- možnosti:
 - Teoretická cesta (povídání o složitosti a těžkosti),
 - praktická cesta (moderní programování – Delphi nebo povídání o jazycích z famílie jazyka C).

Předání funkce parametrem

- Různé datové typy porovnáváme různě (string, integer).
- Chceme-li univerzální funkci, která bude třdit cokoli se jí dostane do ruky, potřebujeme předat porovnávající funkci.
- Syntakticky postupujeme tak, že vytvoříme datový typ tuto funkci nesoucí.
- Do proměnné příslušného typu můžeme odpovídající funkci přiřadit a pak můžeme tuto funkci zavolat.
- Syntax: `type jmeno_typ=function (argumenty:typy):navratovy_typ;`
- `var funkcni_promenna:jmeno_typ;`
- `function porovnej(argumenty:typy):navratovy_typ;...`
- `funkcni_promenna:=porovnej;`
- `funkcni_promenna(parametry);`

Příklad

```
type porovfce=function (a,b:integer):boolean;
var p:porovfce;
function cmp(a,b:integer):boolean;
begin
    cmp:=(a<b);
end;
procedure por(c:porovfce;a,b:integer);
begin
    if(c(a,b)) then writeln('Prvni vetsi!');
end;
begin
    porovnej:=cmp;
    if(p(10,20)) then writeln('Prvni vetsi');
    por(cmp,10,20);
end
```

Vnější třídění

- Načti co největší blok do operační paměti, sestav haldy a při extract-min místo zakořenění posledního zkus načíst další prvek.
- Je-li tento prvek aspoň tak velký jako poslední prvek, co jsme poslali na výstup, zabubblej ho.
- Je-li tento prvek menší než poslední prvek, co jsme poslali na výstup, nepřidávej ho do haldy, dokonči nad haldou heapsort a založ novou haldy.
- Takto získáme v mezích možností dlouhé setříděné bloky, na které můžeme pustit mergesort.

Vnější třídění s více páskami

- Dnes pro praktické použití už asi passé, ale:
- Stává se, že máme k dispozici pásek více (dnes třeba disků).
- Pak je výhodou mergovat ze všech ostatních na jednu.
- Na každé pásce ovšem máme několik bloků.
- Dojde-li obsah jedné pásky, doběhneme současné mergované "bloky" a začneme mergovat na uvolněnou pásku.
- Jak zorganizovat počty bloků na jednotlivých páskách?
- Zobecněnými Fibonacciho čísly.

Organizace počtu bloků pro mergesort

- Předpokládejme, že máme tři pásy.
- Chceme, aby slévání skončilo jedním dlouhým blokem na jedné pásce":
(0,0,1)
- Tudíž předtím na ostatních dvou páskách muselo být po jednom bloku:
(1,1,0)
- Bezprostředně předtím jsme mergovali ze třetí pásy (protože se vyprázdnila) a museli jsme mergovat někam (BÚNO na druhou). Tedy předtím vypadalo rozmístění bloků takto:
(2,0,1)

- Předtím tudíž výpočet vypadal takto:
(0,2,3)
- Předtím zase takto:
(3,5,0)...
- Tedy vždy jde o dvě po sobě jdoucí Fibonacciho čísla (a třetí je nula).
- Vychází totiž rekurence $a_n = a_{n-1} + a_{n-2}$, obecně
 $a_n = a_{n-1} + \dots + a_{n-k}$.

Medián v lineárním čase

- Jak hledat medián v lineárním čase?
- Ve skutečnosti nebudeme hledat medián, ale k -tý nejmenší prvek.

Medián v lineárním čase

- Rozděl vstup na pětice,
- v každé pětici najdi medián,
- najdi medián mediánů (tedy medián mezi mediány petic),
- rozděl vstup na menší a větší,
- zjisti, zda se k -tý nejmenší nachází mezi většími nebo menšími
- pokračuj s hledáním příslušné hromádky.

Medián lineárně detaily

- Jak najít mediány pětic?
- Hrubou silou (v konstantním čase, opakujeme lineárně-krát).
- Jak najít medián mediánů?
- Rekurzívně (zavolej se na pole mediánů pětic).
- Jak "pokračovat na příslušné hromádce?"
- Rekurzívně (zavolej se buďto na menší hromádku a hledej k -tý nejmenší, nebo máme-li hledat v hromádce větších hodnot budiž l počet prvků na menší hromádce a hledej v hromádce větších $k - l$ -tý nejmenší.

Proč je algoritmus lineární?

Protože:

- mediánů pětic je přibližně pětina délky vstupu,
- hromádka menších čísel stejně jako hromádka větších čísel bude mít velikost aspoň $3/10$,
- tudíž každá z hromádek bude mít též velikost nejvýš $7/10$.
- Zbytek je jen indukce.

Indukce:

Složitost algoritmu vede k rekurenci:

$$T(n) = T\left(\frac{n}{5}\right) + T\left(\frac{7}{10}n\right) + kn.$$

Ukážeme, že existuje l takové, že $T(n) \leq ln$:

$$T(n) \leq \frac{ln}{5} + \frac{7ln}{10} + kn = kn + \frac{9}{10}ln$$

a tedy stačí volit $l \geq 10k$.

Výběrové téma: Složitost VS programování?

Konec

...děkuji za pozornost...

Otázky?