Anotace

- Řídké polynomy a matice,
- údržba paměti svépomocí,
- hashování,
- haldy,
- dynamické programování.

Reprezentace spojovými seznamy

- Chceme reprezentovat řídké polynomy (tedy s málo nenulovými koeficienty). Jak to udělat?
- Pomocí spojových seznamů.
- Vhodný je obousměrný spojový seznam...
- ...a možná i cyklický – a v tom případě s hlavou!
- Sčítání polynomů: Prolezení spojových seznamů (podobně jako merge).
- Násobení polynomů: Potřebujeme lézt oběma směry.
- Hlava je k tomu, abychom poznali, že jsme na konci.

Reprezentace řídkých matic

- Opět máme málo nenulových prvků, matici tedy komprimujeme.
- Možností je mnoho, je třeba zejména přemýšlet (při použití). Například:
- Spojový seznam prvků (uspořádaný v obou dimenzích),
- spojový seznam spojových seznamů (řádky v jednom, sloupce ve druhém),
- "čtvrcení za živa", tedy rozdělíme matici na čtyři čtvrtiny, ty dělíme dále, až je ve "čtvrtině" málo prvků...

Funkce nízké úrovně

- `procedure mark(var p:pointer);` – oznámí vrchol haldy (kam až bylo allokováno)
  nepoužívat (radí kolega Kryl)
- `procedure release(var p:pointer);` – nastaví vrchol haldy (odallokuje všechno, co je nad vrcholem)
  nepoužívat (DTTO)
- Podle všeho ve FPC už nejsou, jsou nebezpečné, je to určitý pokus o garbage collection.
- `function MemAvail: longint;` – vrátí počet dostupných bytů na haldě
  (není ve Free Pascalu od verze 2.0)
- `function MaxAvail: longint;` – vrátí délku nejdelšího volného bloku (největší naalokovatelnou velikost)
  (DTTO)

Funkce nízké úrovně

- `GetMem` – naalokuje paměť, narozdíl od `new` nezkoumá kolik – používat jen v případě nouze (radí F. Klaempfl)
- `FreeMem` – odallokuje paměť naalokovanou pomocí `GetMem` – (DTTO)

Příklad použití GetMem/FreeMem

Vyrobíme pole předem neznámé délky

```
type ppole=^tpole;
     tpole=array[1..10000] of longint;
var pole:ppole;
begin
      GetMem(pole,500);{Urafni 500 bytu}
      pole^[10]:=1000;{To je OK}
      pole^[500]:=1024;{To je K. O., pole je male!}
      FreeMem(pole,500);{Melo by stacit i jen
FreeMem(pole);}
end.
```

Textové a binární soubory

Zhodnocení situace

- V zimě jsme se učili pracovat s textovými soubory.
- Proměnnou typu `text` jsme napojili na soubor, otevřeli jsme, četli, zapisovali a zavřeli.
- Široká použitelnost, občas ale chceme naprogramovat například databázi (knih v knihovně).
- K tomu by se hodil soubor úplně jiných vlastností, ve kterém půjde lépe vyhledávat.
- K tomu přesně lze použít binární soubory.
- Jedná se o soubor sestávající z prvků popsané struktury, které můžeme číst resp. zapisovat.
- Jelikož známe velikost dotyčné struktury, můžeme i vyhledávat.

Technické prostředky

- Většinou se shodují s textovými soubory, nicméně:
- Uděláme proměnnou typu `file of nacitany_typ`.
  Například: `var f:file of nesmysl;`
- Funkce `assign`, `reset`, `rewrite`, `read`, `write` a `close` fungují úplně stejně (tedy jako první argument jim předáme příslušnou proměnnou typu `file`.
- Pozor na `append`!
- Hlavní rozdíl:
  - `filesize` – zjistí počet záznamů v souboru,
  - `seek` – nastaví ukazatel na příslušné místo.

Příklad

telefonní seznam

```
type zaznam=record
       jmeno:string[100];
       cislo:string[20];
end;
var f:file of zaznam;
```

Příklad přidání

do binárního souboru

```
procedure pridej;
var zazn:zaznam;
begin readln(zazn.jmeno);
      readln(zazn.cislo);
      assign(f,'databaze.bin');
      {$I-}
      reset(f);
      {$I+}
      if IOResult<>0 then
            rewrite(F);
      seek(f,filesize(f));
      write(f,zazn);
      close(f);
end;
```

Výpis souboru

```
procedure vypis;
var i:integer;
    zazn:zaznam;
begin
      assign(f,'databaze.bin');
      reset(f);
      for i:=1 to filesize(f) do
      begin
            read(f,zazn);
            writeln('Jmeno: ',zazn.jmeno,', cislo:
',zazn.cislo);
      end;
      close(f);
end;
```

Mazází v binárním souboru

Funkce `truncate`

- Zbytek binárního souboru můžeme smazat pomocí funkce `truncate`.
- Jako parametr jí předáme proměnnou typu file (napojenou na nějaký soubor).
- Příklad:
  ```
  reset(f);
  truncate(f);
  ```
  smaže dosavadní obsah souboru
- podobně jako by udělalo `rewrite(f)`, ovšem pro neexistující soubor by to nový nezaložilo!
- Jak zrušit jeden záznam?
- Přepsat ho posledním a poslední zrušit.