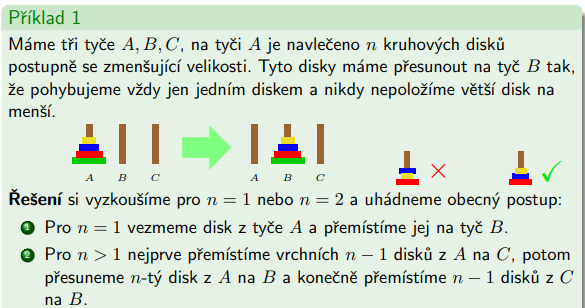
**BI-SPOL-32 Metody řešení rekurentních rovnic, sestavování a řešení rekurentních rovnic při analýze časové složitosti algoritmů**

BI-ZDM

### Rekurence

Obsah obrázku text

Popis byl vytvořen automaticky

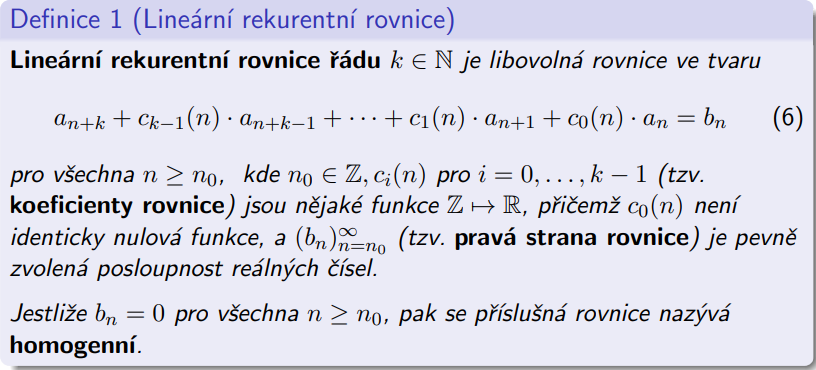


* Potřebujeme kroků

## Metody řešení rekurentních rovnic

### Lineární rekurentní rovnice

**Definice**



* c(n) jsou funkce, mohou to být ale jen konstanty, pak je to LRR s konstantními koeficienty (pak c leží v R a c0 != 0)
* počáteční podmínky jsou, když () až () mají pevně zvolená čísla
* každá LRR má řešení. Pokud má počáteční podmínky, tak má právě jedno řešení

**Diference**

Obsah obrázku text

Popis byl vytvořen automaticky

**Diferenční rovnice**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Řešení rekurentní rovnice**

* řešením je posloupnost an taková, že když dosadíme do rovnice, dostaneme pravdivý výrok pro všechna n
* **obecné řešení** – an = partikulární řešení + řeš. přidružené homogenní rovnice

**Obsah obrázku text

Popis byl vytvořen automaticky**

?? jak se přišlo na tu posloupnost?

**Počáteční podmínky**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Jednoznačnost**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Struktura řešení**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Obsah obrázku text

Popis byl vytvořen automaticky**

### Lineární rekurentní rovnice s konstantními koeficienty (LRRsKK)

**Definice**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Charakteristický polynom**

**Obsah obrázku text

Popis byl vytvořen automaticky**

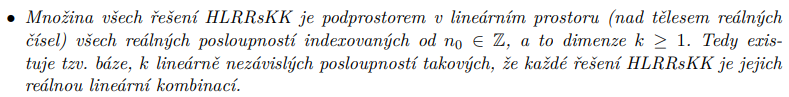
**Řešení rovnice s konstantními koeficienty**

**Obsah obrázku text

Popis byl vytvořen automaticky**

**Obsah obrázku text

Popis byl vytvořen automaticky**

****

**Obsah obrázku text

Popis byl vytvořen automaticky**

= -1 = => -1 = 2\*u + 2 - 4\*u => -3 = -2u => u = 3/2

=> v = 2 - 4\*u => v = -4

**Násobnost charakteristického čísla**

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

**Komplexní charakteristická čísla**

* co dělat, když se nám sejdou charakteristická čísla, která nejsou reálná?

Obsah obrázku text

Popis byl vytvořen automaticky

Nebo an- může být sin(n\*Pí/2) a an~ může být cos(n\*Pí/2) – přišlo se na to tak, že se za první a druhý prvek zvolilo 01 a 10 (báze) a zbytek se dopocital – od toho se odhadl vzorec

**Komplexní charakteristická čísla obecně**

* komplexní číslo má souřadnice x a y v rovině. Toto číslo můžeme vyjádřit pomocí sin a cos – uděláme kolmici k ose x a od [0,0] uděláme přímku k bodu – souřadnici x a y můžeme vyjádřit jako r(přímka k bodu/přepona) \* cos (pro x) a r \* sin (pro y)
* při mocnění – mocníme vzdálenost a násobíme úhel

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

**Příklad HLRRsKK**

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

### Nehomogenní rovnice

* řešení nehomogenní rovnice najdeme pomocí kvalifikovaného odhadu podle tvaru pravé strany

**Kvazipolynom**

Obsah obrázku text

Popis byl vytvořen automaticky

Př.



**Řešení**

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

* an = partikulární řeš. + homo. řeš.
* Když bude na pravé straně chybět tak dosadím
* Další příklad v

**Odhady**

**Obsah obrázku stůl

Popis byl vytvořen automaticky**

**Princip superpozice**

**Obsah obrázku text

Popis byl vytvořen automaticky**

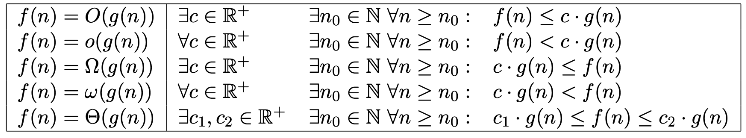
## Sestavování a řešení rekurentních rovnic při analýze časové složitosti

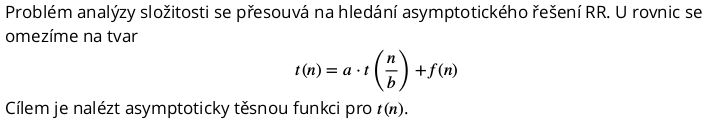
**Analýza složitosti** – výpočet / odhad / predikce požadavků na výpočetní prostředky, které provedení algoritmu bude vyžadovat v závislosti na velikosti vstupních dat

* Složitost algoritmu vyjadřujeme v nejlepším případě, nejhorším případě a v průměrném případě (nejobtížnější)
* Složitost chceme pouze řádově
* O notace – maximálně dosahuje tohoto
* Ω notace – minimálně je složitost tato

**Asymptotická složitost**

* chování funkce pro n →oo, zanedbáváme vše, kromě nejvyššího řádu funkce

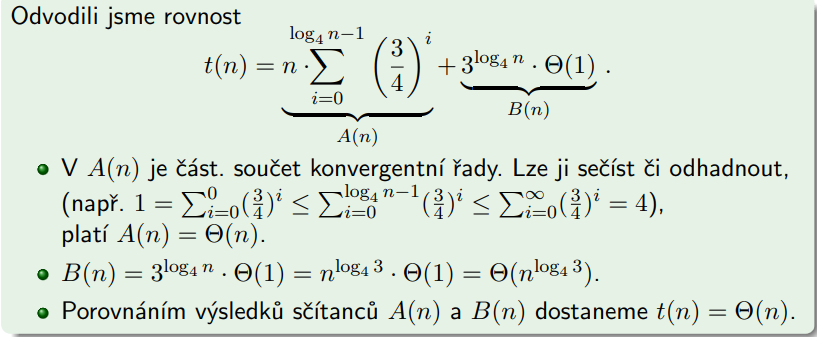
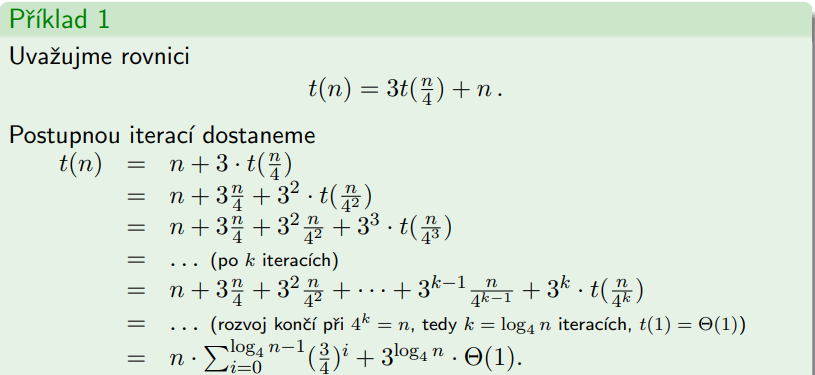




a > 0, b > 1 a f(n) je kladná funkce, uvažujeme pouze horní meze výrazů typu n/b

### Iterační metoda

* rekurentní rovnici expandujeme její iterací, vedoucí na rozvoj konečné řady. Řešením je její součet. Rozvádím tu rekurenci tak daleko, doku v tom neuvidím nějakou sumu
* lze použít pro určení asymptotické horní a dolní meze  
  př.
* základní krok t(1) trvá konstantní dobu Θ(1)
  + odvodíme nějakou číselnou řadu, zůstane nějaký rekurentní zbytek, t(n) musí skončit konstantou -> t(1), spočítáme k (exponent). Pak porovnáváme, co roste rychleji.
  + Ta řada běží až do k – proto si to k musíme vyjádřit, abychom věděli, co dát do řady
  + + · Θ(1) – na konci rozvoje mám 3nak problémů velikosti 1
  + = , protože = =

****

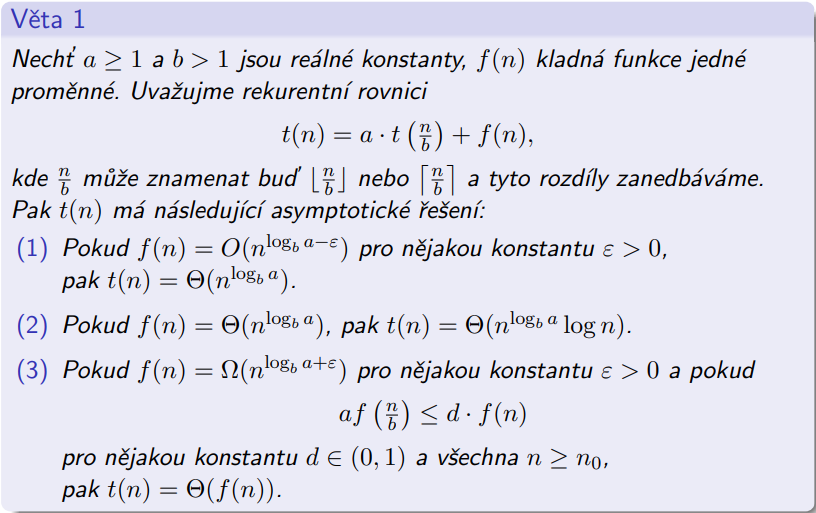
**Obsah obrázku text

Popis byl vytvořen automaticky**

* pro vizualizaci iterační metody může být užitečný strom rekurzivních volání, kde se určuje jeho hloubka, šířka v hloubce *i* a složitost podúlohy v hloubce *i*

### Mistrovská metoda

* odvozena od iterační – aplikujeme iterační metodu na obecnou rekurenci



Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

### Substituční metoda

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automatickyObsah obrázku text

Popis byl vytvořen automatickyObsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky

## Otázky a odpovědi

1. Proč je množina všech řešení LRR vektorový prostor dimenze k?  
   Obsah obrázku text

   Popis byl vytvořen automaticky
2. Obsah obrázku text

   Popis byl vytvořen automatickyJak si představit LRR?
3. K čemu se používá diference?
4. K čemu je princip superpozice?
5. Proč se omezujeme na tvar t(n) = a \* t (n / b) + f(n),
6. Příklad na NLRRsKKaKPPS.
7. Příklady na odhad.

Obsah obrázku text

Popis byl vytvořen automaticky

Obsah obrázku text

Popis byl vytvořen automaticky