Cvičení
14.5.2019

Př: Máme jazyk L, který se skládá z neorientovaných grafů, ve kterých je podgraf trojúhelník. Dále máme algoritmus o 2 krocích:

1) náhodně vybereme jednu hranu $\{x,y\} \in E$ a jeden vrchol $z \in V$, $z \neq x$, $z \neq y$.
2) zkontrolujeme, zda $\{x,z\}$ a $\{y,z\}$ jsou hrany. Pokud ano, úspěšně skončíme.

Kolikrát musíme kroky 1 a 2 opakovat, aby byl algoritmus typu Monte Carlo?

- pravděpodobnost, že jeden běh najde trojúhelník, je $\underbrace{\frac{3}{m}}_{\text{jeden ze tří hran }\Delta\text{, která je v }E} \cdot \underbrace{\frac{1}{n-2}}_{\text{jeden z vrcholů }\Delta\text{, ale ne ten z hrany}}$
- pravděpodobnost neúspěchu je $1 - \frac{3}{m} \cdot \frac{1}{n-2}$
- pravděpodobnost neúspěchu po k opakováních je $\left(1 - \frac{3}{m} \cdot \frac{1}{n-2}\right)^k$

Chceme, aby pravděpodobnost úspěchu po k opakováních byla aspoň 1/2.

$$1 - \left(1 - \frac{3}{m} \cdot \frac{1}{n-2}\right)^k \geq 1/2$$
$$\left(1 - \frac{3}{m} \cdot \frac{1}{n-2}\right)^k \leq 1/2$$
← teď bychom mohli logaritmovat, ale to je zákeřné, uděláme to jinak

Označme $x = \frac{3}{m} \cdot \frac{1}{n-2} = \frac{3}{m(n-2)}$, x je docela malé číslo, $(1-x)^k \simeq e^{\cdot}$

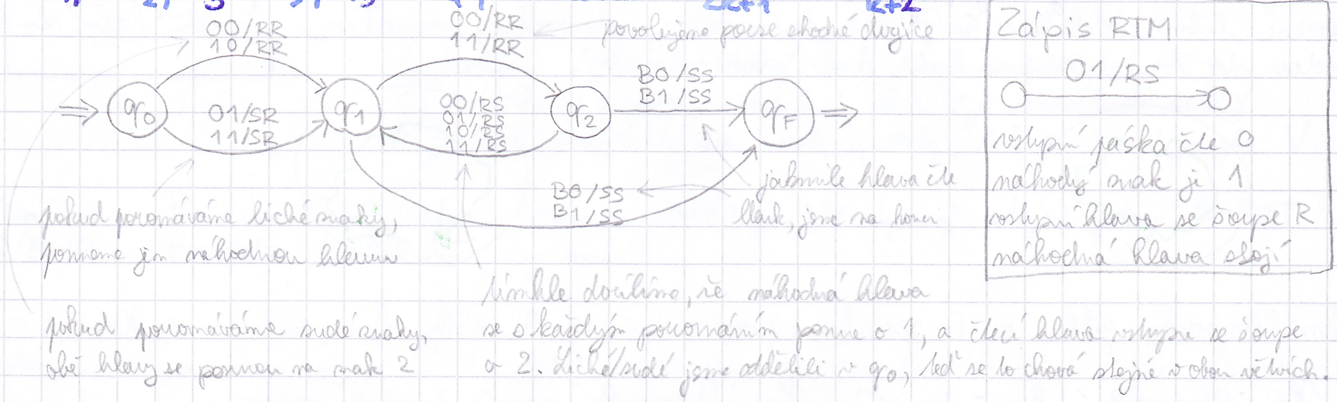
Přijdeme na to přes definici e: $e = \lim_{n \to \infty} \left(1 + \frac{1}{n}\right)^n = \lim_{x \to 0} (1+x)^{\frac{1}{x}} = \lim_{x \to 0} (1-x)^{-\frac{1}{x}}$

Z toho: $\lim_{x \to 0} (1-x)^{\frac{k}{x}} = e^{-k}$, proto $(1-x)^k \simeq e^{-kx}$, $e^{-kx} \leq e^{-1} \leq \frac{1}{2}$

Stačí zvolit $k \cdot x = 1$, tedy $k = \frac{1}{x} = \frac{m \cdot (n-2)}{3}$

Př: Navrhněte RTM. Označme vstupní slovo $w = a_1, a_2, \dots$ a slovo na náhodné pásce: $b_1, b_2, \dots$. $\Sigma = \{0, 1\}$. RTM má mít následující chování:

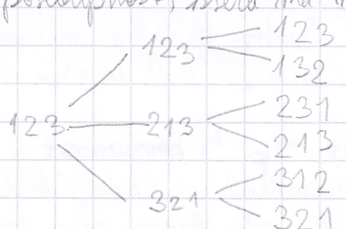
1) pokud $b_1 = 0$, tak porovnáváme sudé znaky w s náhodnou páskou
   $a_2 = b_2$, $a_4 = b_3$, $a_6 = b_4$, obecně: $a_{2k} = b_{k+1}$
2) pokud $b_1 = 1$, tak porovnáváme liché znaky w s náhodnou páskou
   $a_1 = b_2$, $a_3 = b_3$, $a_5 = b_4$, obecně: $a_{2k+1} = b_{k+2}$



Př: Rozhodněte, zda algoritmy generují náhodnou posloupnost

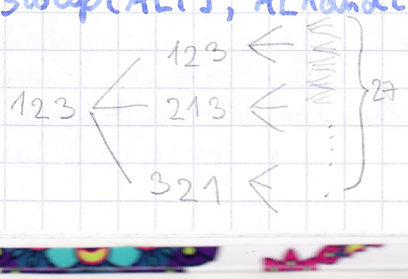
1)
```
For i in range(0, n):
    A[i] = i
For i in range(0, n-1):
    swap(A[i], A[rand(i+1, n)])
```
NE. Protože v každém kroku prohazuji, nikdy mi nemůže vyjít posloupnost, která má na začátku nulu.

2)
```
For i in range (0, n):
    swap(A[i], A[rand(i, n)])
```
ANO, příklad pro n = 3



vygenerovali jsme 6 permutací, všechny se stejnou prstí. tenhle alg. je ok

3)
```
For i in range (0, n):
    swap(A[i], A[rand(1, n)])
```

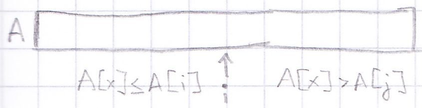


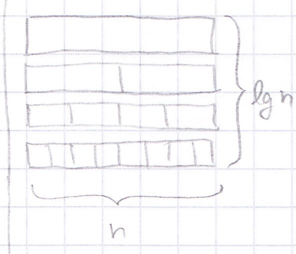
Celkem se vygeneruje 27 různých posloupností, všechny se stejnou pravděpodobností. Na n=3 ale můžeme mít 6 různých posloupností. Protože 27 není dělitelné šesti, je jasné, že některé posloupnosti vzniknou častěji než jiné.

Př: Quick sort: Opakování algoritmu, složitost atd

```
QUICKSORT(F, l):
    if (F < l):
        i = DIVIDE(A, f, l)
        QUICKSORT(f, i-1)
        QUICKSORT(i+1, l)

DIVIDE(A, f, l):
    x = A[l]
    i = f - 1
    for j in range(F, l):
        if A[j] ≤ x
            i += 1
            swap(A[i], A[j])
    return i
```





Složitost: $T(n) = c\cdot n + \frac{1}{n}\cdot\sum_{i=1}^{n}\left(T(i-1) + T(n-i)\right) = c\cdot n + \frac{1}{n}\left(\sum_{i=1}^{n} T(\overbrace{i-1}^{\text{označme } j}) + \sum_{i=1}^{n} T(\overbrace{n-i}^{\text{označme } j})\right) =$

$= cn + \frac{1}{n}\cdot\left(\sum_{j=0}^{n-1} T(j) + \sum_{j=0}^{n-1} T(j)\right) = cn + \frac{2}{n}\cdot\sum_{j=0}^{n-1} T(j)$

Tvrzení: pro $n \geq 2$ platí $T(n) \leq 2c\cdot n\cdot \ln(n)$

Dk silnou indukcí:

$T(0) = T(1) = 0$

1) $n = 2$, pak $T(2) = 2c + \frac{2}{2}\cdot\sum_{j=0}^{1} T(j) = 2c + T(0) + T(1) = 2c + 0 = 2c = k$

$2c\cdot 2\cdot \ln(2) = 4c\cdot\ln(2)$ $\quad 4c\cdot\ln(2) \geq 2c$, ano, platí

2) předpokládám, že tvrzení platí pro $\forall m,\ 2 \leq m \leq n$

$T(n) = cn + \frac{2}{n}\sum_{j=0}^{n-1} T(j) = cn + \frac{2}{n}\left(T(0) + T(1) + \sum_{j=2}^{n-1} T(j)\right) \leq cn + \frac{2}{n}\left(T(0) + T(1)\right) \cdot\cdot$

$+ \frac{4c}{n}\cdot\sum_{j=2}^{n-1} j\cdot\ln(j)$

$\sum_{j=2}^{n-1} j\cdot\ln(j) \leq \int_2^n x\cdot\ln(x)\,dx = \left| \begin{matrix} u = \ln(x) & v' = x \\ u' = \frac{1}{x} & v = \frac{x^2}{2} \end{matrix} \right| = \int_2^n u\cdot v\,dx = \int_2^n \frac{x^2}{2}\cdot\ln(x)dx =$

$= \left[\frac{x^2}{2}\cdot\ln(x)\right]_2^n - \int_2^n \frac{1}{x}\cdot\frac{x^2}{2}\,dx = \left[\frac{x^2}{2}\cdot\ln(x) - \frac{x^2}{4}\right]_2^n = \frac{n^2}{2}\ln(n) - \frac{n^2}{4} - (2\ln(2) - 1) \leq$

$\leq \frac{n^2}{2}\ln(n) - \frac{n^2}{4}$

Proto v indukci použijeme $cn + \frac{4c}{n}\sum_{j=2}^{n-1} j\cdot\ln(n) \leq cn + \frac{4c}{n}\left(\frac{n^2}{2}\ln(n) - \frac{n^2}{4}\right) =$

$= cn + 2c\cdot n\cdot\ln(n) - cn = 2c\cdot n\cdot\lg(n)$

Při vhodné volbě pivotu je třeba $n\cdot\lg n$ kroků, při špatné volbě n^2 kroků. Asymptoticky to ale vychází $n\cdot\lg(n)$ vždycky.