Randomizované algoritmy

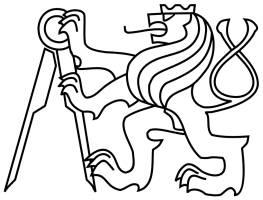
Karel Richta a kol.

Přednášky byly připraveny s pomocí materiálů, které vyrobili Marko Berezovský, Petr Felkel, Josef Kolář, Michal Píše a Pavel Tvrdík

Katedra počítačů
Fakulta elektrotechnická
České vysoké učení technické v Praze

Datové struktury a algoritmy, B6B36DSA
02/2024, Lekce 5

https://cw.fel.cvut.cz/wiki/courses/b6b36dsa/start



Příklad: Výběr zaměstnance

```
HIRE-ASSISTANT(n)
1  best = 0          // candidate 0 is a least-qualified dummy candidate
2  for i = 1 to n
3      interview candidate i
4      if candidate i is better than candidate best
5          best = i
6          hire candidate i
```

Náklady na výběr

- Naším úkolem je spočítat náklady na výběr nejlepšího kandidáta, čas zde není rozhodující.
- Cena za provedení interview: c_i
- Cena za najmutí jednoho kandidáta : c_h
- Celkový počet kandidátů: n
- Najmeme: m
- Obecně jsou náklady: $\boldsymbol{c_i n + c_h m}$
- Nejhorší případ: $\boldsymbol{(c_i + c_h)n}$
- Nejlepší případ: $\boldsymbol{c_i n + c_h}$
- Co se stane v průměrném případě?

Pravděpodobnostní analýza

- **Pravděpodobnostní analýza** – využití znalostí z teorie pravděpodobnosti při analýze složitosti algoritmu.
- Potřebujeme odhad pravděpodobné **distribuce vstupních dat**.
- Průměr za všechny možné vstupy dle distribuce = **průměrný odhad ceny** (nebo času/paměti).

Průměrné náklady

- Problém najmutí zaměstnance:
  - umíme porovnat libovolné dva kandidáty,
  - Označme kandidáty $i = 1..n$ a jejich schopnosti $rank(i)$ – vyšší „*rank*“ znamená lepší schopnosti.
- Stačí nám posloupnost: $\langle rank(1), \ldots, rank(n) \rangle$
- = permutace kandidátů: ***n!*** možností
- Každá je stejně pravděpodobná: **rovnoměrné (uniformní) rozdělení**

Náhodný jev, veličina a pravděpodobnost

- **Jevový prostor:** množina náhodných jevů, př. „hod kostkou“:
  S = {⚀, ⚁, ⚂, ⚃, ⚄, ⚅}
- **Náhodný jev** A je podmnožinou S ($A \subseteq S$):
  A = {⚀, ⚁, ⚂} – interpretace: “maximální hodnota hodu 3“
- **Elementární jev s** je prvek S:
  s = ⚀
- **Náhodná veličina** X přiřazuje náhodnému jevu reálné číslo:
  A_1 = {⚀} ⇒ X=1, A_2 = {⚁} ⇒ X=2, A_3 = {⚂} ⇒ X=3, atd.
- Axiomy **pravděpodobnosti**:
  - $Pr\{A\} \geq 0$ pro $\forall A$
  - $Pr\{S\} = 1$
  - $Pr\{A \cup B\} = Pr\{A\} + Pr\{B\}$ pro $A \cap B = \emptyset$

Střední hodnota náhodné veličiny

- Nejjednodušší a nejlépe použitelný odhad distribuce hodnot náhodné proměnné je “průměr” hodnot, kterých nabývá.
- **Střední hodnota** (očekávání, průměr) E[*X*] diskrétní náhodné proměnné *X* je:

$$E[X] = \sum_{x=1}^{n} x \cdot Pr\{X = x\}$$

- Tato suma je dobře definovaná, jestliže je konečná, nebo konverguje.
- Někdy se očekávaná hodnota *X* zapisuje μ_X (pokud je náhodná proměnná *X* zřejmá z kontextu, stačí μ).
- Věta o **linearitě střední hodnoty**:

  E[*X* + *Y*] = E[*X*] + E[*Y*]

- **Linearita střední hodnoty platí, i když *X* a *Y* jsou závislé ☺**
  - tj. Pr{X ≤*x* ∩ *Y* ≤*y*} = Pr{X ≤*x*} · Pr{Y ≤*y*} pro ∀x a y

Příklad

- Uvažme hru, kdy házíme dvěma mincemi. Výhra je $3 za každého orla, ale -$2 za každou pannu.
- Očekávaná hodnota náhodné veličiny X reprezentující výhru:

$$E[X] = 6 \cdot Pr\{2 \text{ x orel}\} + 1 \cdot Pr\{1 \text{ x orel, } 1 \text{ x panna}\} - 4 \cdot Pr\{2 \text{ x panna}\} =$$

$$= 6 \cdot \tfrac{1}{4} + 1 \cdot \tfrac{1}{2} - 4 \cdot \tfrac{1}{4} = 1$$

Indikátory náhodných veličin

- Předpokládejme, že máme dán jevový prostor S a náhodný jev A. Pak **indikátor náhodné veličiny** I{A} je určen:

$$I\{A\} = \begin{cases} 1 & \text{– pokud A nastane} \\ 0 & \text{– pokud A nenastane} \end{cases}$$

- Uvažme příklad házení mincí – jevový prostor je S = {orel,panna} s pravděpodobnostmi: Pr{orel} = Pr{panna} = ½ . Můžeme definovat indikátor náhodné proměnné X_{orel}, přiřazený k události, kdy padne orel – pak pro daný hod nabyde hodnoty 1, jinak 0.

$$X_{orel} = I\{orel\} = \begin{cases} 1 & \text{– pokud padne orel} \\ 0 & \text{– pokud padne panna} \end{cases}$$

- Očekávaný počet, kolikrát padne orel pro danou hru, je prostě hodnota indikátorové proměnné X_{orel}.

Indikátory a pravděpodobnost

Střední hodnota náhodné veličiny “kolikrát padne orel“:

$$E[X_{orel}] = E[I\{X_{orel}\}] = 1 \times Pr\{orel\} + 0 \times Pr\{panna\} =$$
$$= 1 \times ½ + 0 \times ½ = ½$$

Je to tedy ½ . Lze ukázat, že odpovídá pravděpodobnosti indikované události.

Lemma: Buď prostor *S* možných hodnot a mějme událost *A* z prostoru *S*.
Nechť $X_A = I\{A\}$. *Pak* $E[A] = Pr\{A\}$.

Důkaz: $E[A] = E[I\{A\}] = 1 \times Pr\{A\} + 0 \times Pr\{\neg A\} = Pr\{A\}$

kde $\neg A$ je komplement *A*, tj. $\neg A = S - A$. q.e.d.

Pro *n* hodů mincí, buď $X_i = I$\{výsledek *i*-tého hodu\}. Pak:

$$X = \sum_{i=1}^{n} X_i$$

A očekávaný počet hodů, kdy padne orel:

$$\mathrm{E}[X] = \mathrm{E}[\sum_{i=1}^{n} X_i] = \sum_{i=1}^{n} \mathrm{E}[X_i] = \sum_{i=1}^{n} \frac{1}{2} = n/2$$

Zaměstnanci: analýza průměrných nákladů

$$\mathrm{E}[X] = \sum_{x=1}^{n} x \Pr\{X = x\}$$

$$\begin{aligned} X_i &= \mathrm{I}\{\text{candidate } i \text{ is hired}\} \\ &= \begin{cases} 1 & \text{if candidate } i \text{ is hired}, \\ 0 & \text{if candidate } i \text{ is not hired} \end{cases} \end{aligned}$$

$$X = X_1 + X_2 + \cdots + X_n$$

Zaměstnanci: analýza průměrných nákladů

- Platí:

$$\mathrm{E}[X_i] = \Pr\{\text{candidate } i \text{ is hired}\}$$

- Kandidáta *i* najmeme pouze když je lepší, než kandidáti *1..i-1*.
- Předpokládáme, že kandidáti přicházejí v náhodném pořadí.
- Pravděpodobnost toho, že je *i* je nejlepší z kandidátů *1..i* je: $\mathrm{E}[X_i] = 1/i$
- Platí tedy:

$$\mathrm{E}[X] = \mathrm{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathrm{E}[X_i] = \sum_{i=1}^{n} 1/i$$

$$= \ln n + O(1)$$

součet harmonické řady

$$\lg n \leq \sum_{k=1}^{n} \frac{1}{k} \leq \lg n + 1$$

- Průměrné náklady tedy činí: $O(c_i \times n + c_h \times \ln n)$.
- V nejhorším případě: $O(c_i \times n + c_h \times n)$.

Pravděpodobnostní analýza a randomizované algoritmy

- Algoritmus je **deterministický** – pro jakýkoliv konkrétní vstup je výstup vždy stejný.
- Najímání kandidátů:
  - A1 = <1, 2, 3, 4, 5, 6, 7, 8, 9, 10>, najímáme 10x,
  - A2 = <10, 9, 8, 7, 6, 5, 4, 3, 2, 1>, najímáme 1x,
  - A3 = <5, 2, 1, 8, 4, 7, 10, 9, 3, 6>, najímáme 3x.
  - Cena závisí na počtu najmutí, vidíme, že existují drahé vstupy (A1), ale i levné (A2) a středně obtížné (A3).
- Změna na **randomizovaný algoritmus:**
  - nejprve permutuje vstup a pak vybírá kandidáty,
  - nejsme závislí na vstupní distribuci, ale stále můžeme generovat "špatnou permutaci" (A1).

Formální model randomizace

- RAM model obohatíme o novou instrukci:
  - *randInt(C)* vracející náhodné celé číslo z intervalu mezi *0* a *C−1*.
- Cena za náhodný výběr je **jedna časová jednotka**, jako u všech ostatních instrukcí.
- Algoritmy (programy), které ji nepoužijí jsou **deterministické**, ty které ji použijí jsou **randomizované/stochastické.**
- Čas běhu **randomizovaného** algoritmu je obecně **náhodná veličina**.
- Konečnost můžeme garantovat použitím časovače: po plynutí času *t(n)* se běh zastaví (což však negarantuje správnost výstupu).

Permutace řazením

```
PERMUTE-BY-SORTING(A)
1  n = A.length
2  let P[1..n] be a new array
3  for i = 1 to n
4      P[i] = RANDOM(1, n^3)
5  sort A, using P as sort keys
```

- Volíme interval *1…n^3* aby bylo pravděpodobné (s pravděpodobností alespoň *1-1/n*), že všechny priority v *P* jsou unikátní.
- Lze ukázat, že permutace tříděním generuje **rovnoměrné rozdělení**.
- Nejtěžší krok v permutaci je setřídění pole *A*. Dolní mez pro třídění porovnáním prvků je $O(n \ln n)$, ale zde můžeme použít i metody, které nejsou založené na porovnávání a řadí v čase $O(n)$.

Permutace prohazováním Fisher-Yates/Knuth

```
RANDOMIZE-IN-PLACE(A)
1  n = A.length
2  for i = 1 to n
3      swap A[i] with A[RANDOM(i, n)]
```

Pomocí invariantu: pro každou permutaci prvků vstupního pole platí, že na začátku $i$té iterace cyklu obsahuje pole A na indexech 1 až $i-1$ prvních $i-1$ prvků této permutace s pravděpodobností $(n-i+1)!/n!$.

Randomizované hledání zaměstnance

```
RANDOMIZED-HIRE-ASSISTANT(n)
1  randomly permute the list of candidates
2  best = 0          // candidate 0 is a least-qualified dummy candidate
3  for i = 1 to n
4      interview candidate i
5      if candidate i is better than candidate best
6          best = i
7          hire candidate i
```

- Střední hodnota ceny hledání je opět: $O(c_i \times n + c_h \times \ln n)$.
- V nejhorším případě opět: $O(c_i \times n + c_h \times n)$.
- Ale **zbavili jsme se závislosti na agentuře**.
- Odpovídá problému počítání maxim **zleva-do-prava**.

Příklad: Výběr ceny

- Předpokládejte, že máte šanci účastnit se TV show.
- Je tam **100 skříněk**, které budete otevírat v pořadí, které si stanovíte Vy, **moderátor Vám může napovídat**.
- Skříňka s číslem *i* obsahuje m_i peněz (**hodnota**).
  Nevíte, kolik to je, ale po otevření to zjistíte.
- Žádné dvě skříňky neobsahují stejné množství peněz.
- Pravidla hry jsou jednoduchá:
  - Na začátku hry dostanete **10 žetonů**.
  - Když otevřete skříňku, která má vetší hodnotu, než jakákoliv z dosud otevřených skříněk, musíte **vrátit 1 žeton**.
  - Když musíte vrátit žeton a nemáte ho – prohráli jste.
  - Když se Vám podaří otevřít všechny skříňky, vyhrál jste a můžete si ponechat všechny peníze.

Příklad: Výběr ceny (pokr.)

- Analyzujme nejprve případ, kdy vždy dáte na radu moderátora – jeho nápověda bude řídit hru.
- **Nejhorší případ** bude, když Vás nechá otvírat schránky v pořadí rostoucí hodnoty. Kdykoliv otevřete skříňku, musíte vrátit žeton a po otevření 11-té skříňky prohráváte. To by se ale nelíbilo divákům a proto to asi takto nebude.
- **Nejlepší případ** by byl, kdyby Vám hned prozradil nejdražší skříňku. To by se líbilo hráči, ale nebyl by prostor pro reklamy během hry, a proto to asi takto také nebude.
- Problém lze přirovnat k **hledáním maxima zleva-do-prava**.
- Kdykoliv narazíme na maximum zleva-do-prava, musíme vrátit žeton.
- Pro $n=100$, $H_{100} \leq 1 + \ln 100 = 1 + 4.61$ a $H_{100} < 6$, kde $H_n = \sum_{1 \leq k \leq n} 1/k$ je tzv. „n-tá harmonická“.
- Pokud otevíráte skříňky **v náhodném pořadí**, měli byste v průměru **vrátit méně než 6 žetonů a vyhrát**.

Příklad: Výběr ceny (pokr.)

- Moderátor po Vaší výhře netouží – přesněji netouží po mnoha výhercích, občas ale musí někoho nechat vyhrát – mohli byste to být právě Vy.
- Řešení: **ignorovat nápovědu moderátora a otevírat skříňky v náhodném pořadí**.
- Jak náhodně vybrat skříňku? Pokud zbývá k skříněk, hoďte si kostkou s k stranami a dostanete číslo v intervalu $1..k$.
- Toto je r**andomizovaný algoritmus**. V případě randomizovaného algoritmu jistotu výhry nemáte, ale lépe to nejde.

Příklad: Výběr ceny (pokr.)

- Očekávaný počet vrácených žetonů je méně než 6.
- **Jak si můžete být jisti, že odejdete s výhrou?**
- Spočítáme pravděpodobnost, že počet maxim zleva-doprava (M_n) ≥ 11.
- **Markovova nerovnost** nám umožňuje omezit tuto pravděpodobnost. Říká, že pro nezápornou náhodnou veličinu *X* a libovolnou konstantu $c \geq 1$, platí, že:

  $$Pr\{X \geq c \cdot E[X]\} \leq 1/c.$$

- Pro $X = M_n$ a $c = 11/6$, dostáváme:

  $$Pr\{M_n \geq 11\} \leq Pr\{M_n \geq \frac{11}{6} E[M_n]\} \leq \frac{6}{11}.$$

- Pravděpodobnost výhry $Pr\{M_n < 11\}$ je proto více než 5/11.

Las Vegas

- **Las Vegas:**
  - varianta randomizovaného algoritmu
  - vždy spočte **správný výsledek** (nebo informuje o chybě),
  - **čas běhu je náhodná veličina** (obecně různý i pro stejný vstup).
  - příklad: vstupem je pole A obsahující půlku 1 a druhou půlku 0:

```
FIND_FIRST_ONE_LV(A):
    while True:
        i = randInt(A.length)
        if A[i] == 1:
            return i
```

  - střední hodnota počtu iteraci je $\lim_{n\to\infty} \sum_{i=1}^{n} \frac{i}{2^i} = 2$.

Monte Carlo

- **Monte Carlo**:
  - **čas běhu je vždy stejný** (deterministický),
  - s (nízkou) pravděpodobností může dávat **nekorektní výsledky**.

```
FIND_FIRST_ONE_MC(A, k):
    for j in 1 to k:
        i = randInt(A.length)
        if A[i] == 1:
            return i
```

  - pravděpodobnost nalezení 1 po k krocích je $1 - (1/2)^k$

Las Vegas ->Monte Carlo

- Převod Las Vegas algoritmu na Monte Carlo?
- **Omezíme čas** běhu na *t(n).*
- Pokud **stihne dopočítat**, vrátíme výsledek.
- Pokud ne, vrátíme **náhodnou odpověď**.
- Jak zvolit *t(n)*?
  - *X* je doba běhu Las Vegas (náhodná veličina).
  - Umíme spočítat *E[X].*
  - Zvolme t(n) = $c \cdot E[X]$,
  - použijme Markovovu nerovnost $Pr\{X \geq c \cdot E[X]\} \leq 1/c$:

    $Pr\{X < t(n)\} = 1 - Pr\{X \geq t(n)\} = 1 - Pr\{X \geq c\, E[X]\} \geq 1 - 1/c.$
  - Spodní mez $Pr\{X < t(n)\}$, pravděpodobnosti, že LV stihne dopočítat a tudíž vrátit korektní výsledek, umíme volbou *c* libovolně zvýšit.

Monte Carlo -> Las Vegas

- Jak převést Monte Carlo algoritmus na Las Vegas?
- **Časová složitost** Monte Carlo algoritmu je *m(n)*.
- **Korektní odpověď** vrací s pravděpodobností *p.*
- Máme deterministický algoritmus, který **verifikuje v čase *v(n)***, zda algoritmus Monte Carlo dává korektní odpověď.
- Kombinací lze získat algoritmus typu Las Vegas s očekávaným časem exekuce: $(m(n)+v(n))/(1-p)$.

Příklad: Narozeninový paradox

- Kolik lidí musí být v místnosti, aby byla šance 1:1 (50%), že tam budou nejméně dvě osoby, které se narodily ve stejný den v roce?
- Odpověď je překvapivě nízká - paradox spočívá v tom, že je to mnohem méně, než počet dnů v roce, nebo dokonce než polovina roku.
- Přidělme osobám v místnosti čísla od *1* do *k*.
- Rok má *n = 365* dní.
- b_i je den narození osoby $1 \leq i \leq k$ v roce.
- Předpokládáme uniformní rozdělení: $Pr\{b_i = r\} = 1/n$, kde $i = 1,..,k,\ r = 1,..n$.
- Předpokládáme nezávislost dnů narození.
- Pravděpodobnost, že se dvě různé osoby *i* a *j* narodily ve stejný den *r*:

  $$Pr\{b_i = r \text{ a } b_j = r\} = Pr\{b_i = r\} \cdot Pr\{b_j = r\} = 1/n^2.$$

- Pravděpodobnost, že se narodily ve stejný den:

  $$Pr\{b_i = b_j\} = \sum_{r=1}^{n} Pr\{bi = r \text{ and } bj = r\} = \sum_{r=1}^{n} 1/n^2 = 1/n.$$

Příklad: Narozeninový paradox (pokr.)

- **Chceme**: pravděpodobnost, že nejméně 2 osoby mají stejný den narození.
- **Komplementární tvrzení**: všechna data narození jsou odlišná.
- **Náhodný jev A_i**: osoba *i* má den narození odlišný od osoby *j* pro $\forall j < i$.
- **Náhodný jev B_k**: *k* osob má odlišná data narození: $B_k = \bigcap_{i=1}^{k} A_i$.
  - Můžeme psát $B_k = A_k \cap B_{k-1}$, a z definice podmíněné pravděpodobnosti:

$$Pr\{B_k\} = Pr\{B_{k-1}\}\, Pr\{A_k \mid B_{k-1}\}, \text{ kde triviálně } Pr\{B_1\} = Pr\{A_1\} = 1.$$

- Pokud jsou $b_1, b_2, \ldots, b_{k-1}$ různá data, pak podmíněná pravděpodobnost, že $b_k \neq b_i$, pro i = *1, … , k-1 je* $Pr\{A_k / B_{k-1}\} = (n-k+1)/n$, protože z *n* dnů *se* *n-(k-1)* už nesmí vybrat.

Příklad: Narozeninový paradox (pokr.)

$\Pr\{B_k\} = \Pr\{B_{k-1}\} \Pr\{A_k \mid B_{k-1}\} = \Pr\{B_{k-2}\} \Pr\{A_{k-1} \mid B_{k-2}\} \Pr\{A_k \mid B_{k-1}\} =$

. . .

$= \Pr\{B_1\} \Pr\{A_2 \mid B_1\} \Pr\{A_3 \mid B_2\} \ldots \Pr\{A_k \mid B_{k-1}\} =$

$= 1 \cdot \left(\frac{n-1}{n}\right) \cdot \left(\frac{n-2}{n}\right) \ldots \left(\frac{n-k+1}{n}\right) =$

$= 1 \cdot \left(1 - \frac{1}{n}\right) \cdot \left(1 - \frac{2}{n}\right) \ldots \left(1 - \frac{k-1}{n}\right) \leq$ (protože $e^x \geq 1+x$)

$\leq e^{-1/n} e^{-2/n} \ldots e^{-(k-1)/n} =$

$= e^{-\sum_{i=1}^{k-1} i/n} =$

$= e^{-k(k-1)/2n} \leq$

$\leq 1/2$

kde: $-k(k-1)/2n \leq \ln(1/2)$

Příklad: Narozeninový paradox (pokr.)

- Pravděpodobnost, že budou všechna data narození různá je nejvýše ½, pokud k(k-1) ≥ 2n·ln(2), pokud:
  $k \geq (1+\sqrt{1 + (8 \ln 2)n})/2$
- Pro n = 365 vychází $k \geq 23$.
- Pokud je tedy v místnosti 23 a více lidí, je pravděpodobnost ½ (50%), že alespoň dva lidé budou mít stejné datum narození.

Příklad: Narozeninový paradox s použitím indikátorů

- Pro každý pár *lidí (i, j), 1≤i<j ≤k* definujeme $X_{ij} = I\{bi = b_j\}$
- Pravděpodobnost, že dva lidé mají stejné narozeniny je *1/n*, tedy: $E[X_{ij}] = Pr\{b_i = b_j\} = 1/n$ pro *1≤i<j ≤k*
- Výsledná náhodná veličina je:

$$X = \sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij}$$

- A její střední hodnota:

$$\mathrm{E}[X] = \mathrm{E}\left[\sum_{i=1}^{k} \sum_{j=i+1}^{k} X_{ij}\right] = \sum_{i=1}^{k} \sum_{j=i+1}^{k} \mathrm{E}[X_{ij}] = \binom{k}{2}\frac{1}{n} = \frac{k(k-1)}{2n}$$

- Pro *k(k-1) ≥2n* bude *E[X] ≥1*, pro alespoň $\sqrt{2n} + 1$ lidí můžeme očekávat alespoň jeden pár se stejnými narozeninami. Zvolíme *k = 28*.
- Liší se drobně od předchozího řešení, ale obě jsou $\theta(\sqrt{n})$.

The End