

**Verjetnostne metode v  
računalništvu - zapiski s  
predavanj prof. Marca**

Tomaž Poljanšek

študijsko leto 2023/24

# Kazalo

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Probability . . . . .	1
1.2	Random variables . . . . .	2
<b>2</b>	<b>Quicksort, min-cut</b>	<b>4</b>
2.1	Quicksort . . . . .	4
2.2	Min-cut . . . . .	6
<b>3</b>	<b>Complexity classes</b>	<b>9</b>
<b>4</b>	<b>Chernoff bounds</b>	<b>11</b>
<b>5</b>	<b>Monte Carlo methods</b>	<b>16</b>
5.1	Example 1 . . . . .	16
5.2	Example 2 . . . . .	16
5.3	$(\epsilon, \delta)$ -approximation . . . . .	17
5.4	DNF counting . . . . .	18
<b>6</b>	<b>Polynomials</b>	<b>20</b>
<b>7</b>	<b>Random graphs</b>	<b>24</b>
7.1	$G(n, p)$ model . . . . .	24
7.2	Barbási-Albert Model . . . . .	27
<b>8</b>	<b>Markov chains</b>	<b>29</b>

8.1	2-SAT . . . . .	32
8.2	Generating a uniformly random element of a set . . . . .	33
8.3	Metropolis algorithm . . . . .	34
8.4	M.C. for 1-factor in bipartite graphs . . . . .	37
<b>9</b>	<b>Randomized incremented constructions (RIC)</b>	<b>39</b>
9.1	Quicksort as RIC . . . . .	40
9.2	Linear programming . . . . .	41
<b>10</b>	<b>Hashing</b>	<b>46</b>
10.1	Chaining . . . . .	49
10.2	2 level hashing . . . . .	51
10.3	The power of 2 choices . . . . .	52
10.4	Cockoo hashing . . . . .	53
10.5	Bloom filter . . . . .	55
10.6	Linear probing . . . . .	55
<b>11</b>	<b>Data streams</b>	<b>57</b>
11.1	Count min sketch . . . . .	58

# Poglavje 1

## Introduction

### 1.1 Probability

$(\Omega, F, P_r)$ :

- $\emptyset \in F$ ,
- $A \in F \implies A^c \in F$ ,
- $A_1, A_2 \dots \in F \implies \cup_{i=1}^{\infty} A_i \in F$ .

$P_r(A) \geq 0$ ,

$P_r(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P_r(A_i)$  if  $A_i$  disjoint,

$P_r(\cup_{i=1}^{\infty} A_i) \leq \sum_{i=1}^{\infty} P_r(A_i)$ ,

$\Omega = \{\omega_1, \omega_2 \dots\}$  - countable case.

$$\begin{pmatrix} \omega_1 & \omega_2 & \dots \\ p_1 & p_2 & \dots \end{pmatrix}$$

*Primer.*

`Alg():`

`while True:`

`B = sample as random from {0,1} # 1 with probability p`

`if B = 1:`

return

$$\Omega = \{1, 01, 001, 0001 \dots\}$$

$$\begin{pmatrix} 1 & 01 & 001 & 0001 & \dots \\ p & (1-p)p & (1-p)^2p & (1-p)^3p & \dots \end{pmatrix}.$$

## 1.2 Random variables

$X : \Omega \rightarrow \mathbb{Z}$ .

$E[X] = \sum_{c \in \mathbb{Z}} c \cdot P_r(X = c)$  expected value of  $X$ .

Properties:

- $E[f(X)] = \sum_{c \in \mathbb{Z}} f(c) \cdot P_r(X = c)$ ,
- $E[aX + bY] = aE[X] + bE[Y]$ ,
- $E[X \cdot Y] = E[X] \cdot E[Y]$  if  $X, Y$  independent,
- $P_r(X \geq a) \leq \frac{E[X]}{a} \forall a > 0, X \geq 0$  Markov inequality.

*Primer.* (Continuing from before).

$X$  = number of trials before return.

$X : \Omega \rightarrow \mathbb{Z}$ .

$X : 1 \rightarrow 1, 01 \rightarrow 2, 001 \rightarrow 3 \dots$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \dots \\ p & (1-p)p & (1-p)^2p & (1-p)^3p & \dots \end{pmatrix} - \text{geometric distribution.}$$

**Trditev 1.2.1.**  $E[X] = \frac{1}{p}$ .

**Dokaz 1.2.2.**  $X = \sum_{i=1}^{\infty} X_i$ .

$$X_i = \begin{cases} 1 & \text{if trial } i \text{ is executed} \\ 0 & \text{else} \end{cases}$$

$$\begin{aligned} E[X] &= E\left[\sum_{i=1}^{\infty} X_i\right] = \sum_{i=1}^{\infty} E[X_i] = \\ &= \sum_{i=1}^{\infty} (1-p)^{i-1} = \frac{i=0}{\infty} (1-p)^i = \frac{1}{1-(1-p)} = \frac{1}{p}. \end{aligned}$$

$$E[X] = \frac{1}{p}.$$

$$P_r(X \geq 100 \cdot \frac{1}{p}) \leq \frac{E[X]}{\frac{1}{p}} = \frac{1}{100}.$$

**Definicija 1.2.3.**  $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{i=1}^{\infty} \frac{1}{i}$ .

**Izrek 1.2.4.**  $H_n \leq 1 + \ln(n)$ .

**Dokaz 1.2.5.**

$$H_n = 1 + \sum_{i=2}^n \frac{1}{i} \stackrel{\text{integral}}{\leq} 1 + \int_1^n \frac{dx}{x} = 1 + \ln(x)|_1^n = 1 + \ln(n).$$

# Poglavje 2

## Quicksort, min-cut

### 2.1 Quicksort

Input: set (no equal element) (unordered list)  $S \in \mathbb{R}$   
(or whatever you can compare linearly)

Output: ordered list

Code:

```
def Quicksort(S):  
    if |S| = 0 or 1:  
        return S  
    else:  
        a = uniformly at random from S  
         $S^- = \{b \in S \mid b < a\}$   
         $S^+ = \{b \in S \mid a < b\}$   
        return Quicksort( $S^-$ ), a, Quicksort( $S^+$ )
```

$C(n)$  - random variable, the number of comparisons in evaluation of Quicksort with  $|S| = n$ .

**Izrek 2.1.1.**  $E[C(n)] = O(N \log(n))$ .

**Dokaz 2.1.2.**  $C(0) = C(1) = 0$ .

$$\begin{aligned}
E[C(n)] &= n - 1 + \sum_{i=1}^n (E[C(i-1)] + E[C(n-i)]) \cdot P_r(a \text{ is } i\text{-it element}) \leq \\
&\leq n + \frac{2}{n} \sum_{i=1}^{n-1} E[C(i)].
\end{aligned}$$

Induction:

$n = 1 : \checkmark$

$n - 1 \rightarrow n$ :

$$\begin{aligned}
E[C(n)] &\leq n + \frac{2}{n} \sum_{i=1}^n E[C(i)] \leq \\
&\leq n + \frac{2}{n} \sum_{i=1}^n 5i \log i \leq \\
&\leq n + \frac{2}{n} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 5i \log i + \frac{2}{n} \sum_{i=1+\lfloor \frac{n}{2} \rfloor}^{n-1} 5i \log i \leq \\
&\leq n + \frac{2}{n} \sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} 5i \log \frac{n}{2} + \frac{2}{n} \sum_{i=1+\lfloor \frac{n}{2} \rfloor}^{n-1} 5i \log n \leq \\
&(\log \frac{n}{2} = \log n - 1) \\
&\leq n + \frac{2}{n} \left( \sum_{i=1}^n 5i \log n - \sum_{i=1}^{\frac{n}{2}} 5i \right) = \\
&= n + \frac{10}{n} \left( \frac{n(n-1)}{2} \log n - \frac{\frac{n}{2}(\frac{n}{2}+1)}{2} \right) \leq \\
&\leq n + 5(n-1) \log n - n < \\
&< 5n \log n.
\end{aligned}$$

$$P(C(n) \geq b \cdot 5n \log n) \stackrel{\text{Markov}}{\leq} \frac{1}{b}.$$

**Dokaz 2.1.3.**

2:

Let  $S_1, S_2 \dots S_n$  sorted elements of  $S$ .

Define random variable  $X_{ij} = \begin{cases} 1 & \text{if } S_i \text{ and } S_j \text{ are compared} \\ 0 & \text{else} \end{cases}$



$$C(n) = \sum_{1 \leq i < j \leq n} E[X_{ij}].$$

$$E[X_{ij}] = P(S_i \text{ and } X_j \text{ compared}).$$

$S_{ij}$  - the last set including  $S_i$  and  $S_j$ .

$$E[X_{ij}] = \frac{2}{|S_{ij}|} \leq \frac{2}{j-i+1}.$$

$$|S_{ij}| \geq j - i + 1.$$

$S_{ij}$  has everything in between.

$$\begin{aligned} \Rightarrow E[C(n)] &\leq \sum_{1 \leq i < j \leq n} \frac{2}{j-i+1} = \\ &= \sum_{k=j-i+1}^{n-1} \sum_{i=1}^{n-i+1} \frac{2}{k} \leq \\ &\leq 2 \cdot n \cdot H_n \leq \\ &\leq 2n(1 + \log n). \end{aligned}$$

## 2.2 Min-cut

$G$  multigraph.

Cut:  $U \subset V(G)$ ,  $U \neq \emptyset, V(G)$ .

$$(U, V(G) \setminus U) = \{uv \in E(G) \mid u \in U, v \in V(G) \setminus U\}.$$

Problem min-cut:

Input:  $G$ .

Output:  $\min |(U, V(G) \setminus U)|$  - cut size.

Algorithm 1:

$x \in V(G)$

Call  $\text{maxFlow}(G, x, y) \forall y \in V(G)$

Take  $\min$

$\text{maxFlow}$  is Edmonds-Karp algorithm  $O(|V||E|^2)$ .

Algorithm 2 (Stoer Wagner)

Is  $O(|E||V| + |V|\log|V|)$ .

Algorithm *randMinCut*:

```

G_0 = G
i = 0
while |V(G_i)| > 2:
    e_i = uniformly at random from G_i
    G_{i+1} = G_i / e_i
    i = i + 1
u, v = V(G_{n-2}) // n = |V(G)|
U = {w ∈ V(G) | w is merged into u}
return (U, V(G) \ U)

```

**Izrek 2.2.1.** Algorithm *randMinCut* gives you a minimal cut with probability greater or equal to  $\frac{2}{n(n-1)}$ .

**Dokaz 2.2.2.**

Fact 1:  $\minCut(G_i) \leq \minCut(G_i)$ ;

$\not\leq$ : *minCut* remains.

Fact 2:  $\minCut(G) \leq \delta(G)$ .

$k := \minCut(G)$ .

Let  $(A, B)$  be an optimal cut.

$\epsilon_i$  not in  $(A, B)$ .

$$\begin{aligned}
 & P_r(\text{Algorithm not returning } (A, B)) \\
 &= P_r(\epsilon_0 \cap \dots \cap \epsilon_{n-3}) \\
 &= P_r(\epsilon_0 \cap \dots \cap \epsilon_{n-4}) \cdot P_r(\epsilon_{n-3} \mid \epsilon_0 \cap \dots \cap \epsilon_{n-4}) \\
 &= P_r(\epsilon_{n-3} \mid \cap_{i=0}^{n-4} \epsilon_i) \cdot P_r(\epsilon_{n-3} \mid \cap_{i=0}^{n-4} \epsilon_i) \\
 &\dots P_r(\epsilon_1 \mid \epsilon_0) \cdot P_r(\epsilon_0). (*)
 \end{aligned} \tag{2.1}$$

$$P_r(\bar{\epsilon}_i \mid \epsilon_{i-1} \cap \dots \cap \epsilon_0) = \frac{k}{|E(G_i)|} \stackrel{(**)}{\leq} \frac{k}{\frac{(n-i)k}{2}} = \frac{2}{n-i}$$

$$|E(G_i)| \geq \frac{(n-i)\delta(G)}{2} \geq \frac{(n-i)k}{2}. (**) \tag{2.2}$$

$$P_r(\epsilon_i \mid \epsilon_{i-1} \cap \dots \cap \epsilon_0) \geq 1 - \frac{2}{n-i} = \frac{n-2-i}{n-i}.$$

$$(*) \geq \frac{n-2}{n} \cdot \frac{n-3}{n-1} \cdots \frac{1}{3} = \frac{2}{n(n-1)}.$$

**Izrek 2.2.3.** Running *randMinCut*  $n(n-1)$  times and taking best output gives correct solution with probability  $\geq 0.86$ .

**Dokaz 2.2.4.**  $A_i$  - event that  $i$ -th run gives sub-optimal solution.

$$\begin{aligned} P_r(\text{solution not correct}) &= P_r(A_1 \cap \dots \cap A_{n(n-1)}) \\ &= \prod_{i=1}^{n(n-1)} P_r(A_i) \leq \left(1 - \frac{2}{n(n-1)}\right)^{n(n-1)} \\ &\leq e^{-\frac{2}{n(n-1)} \cdot n(n-1)} = e^{-2} \leq 0.14. \end{aligned}$$

$$1 - x \leq e^x \quad \forall x \in \mathbb{R}.$$

If we run  $n(n-1)\log(n)$  times  $\rightarrow O\left(\frac{1}{n}\right)$ .

$O(n^2 \log n \cdot n)$ .

Improved:  $O(n^2 \log^3 n)$ .

## Poglavje 3

### Complexity classes

Decision problem - yes/no question on a set of inputs = asking  $w \in \Pi$ .

Randomized algorithms:

- Las Vegas algorithms: always gives correct solution, example: *Quicksort*.
- Monte Carlo algorithms: it can give wrong answers. Monte Carlo algorithms subtypes:

$$- \text{type}(1): \begin{cases} \text{if } \omega \in \Pi \implies \text{algorithm returns } „\omega \in \Pi“ \text{ with probability } \geq \frac{1}{2} \\ \text{if } \omega \notin \Pi \implies \text{algorithm returns } „\omega \in \Pi“ \text{ with probability } = 0 \end{cases}$$

$$- \text{type}(2): \begin{cases} \text{if } \omega \in \Pi \implies \text{algorithm returns } „\omega \in \Pi“ \text{ with probability } = 1 \\ \text{if } \omega \notin \Pi \implies \text{algorithm returns } „\omega \in \Pi“ \text{ with probability } \leq \frac{1}{2} \end{cases}$$

$$- \text{type}(3): \begin{cases} \text{if } \omega \in \Pi \implies \text{algorithm returns } „\omega \in \Pi“ \text{ with probability } \geq \frac{3}{4} \\ \text{if } \omega \notin \Pi \implies \text{algorithm returns } „\omega \in \Pi“ \text{ with probability } \leq \frac{1}{2} \end{cases}$$

type(1) and type(2): one-sided error, type(3): 2-sided error.

$\frac{1}{2}$ ,  $\frac{3}{4}$  and  $\frac{1}{4}$  arbitrary numbers, can be something different (for type(3) better than coin flip).

*Primer.* Decisional problem: does a graph  $G$  have  $\text{minCut} \leq k$ ?

Run  $randMinCut(G)$   $n(n-1)$  times.

```
Algorithm randMinCut:
  if one of runs gives  $|A, B| \leq k$ :
    return true
  else:
    return false
```

Complexity classes:

- RP (randomized polynomial time): decisional problems for which there exists Monte Carlo algorithm of type(1) with polynomial time complexity (worst case).
- co-RP: decisional problems for which there exists Monte Carlo algorithm of type(2) with polynomial time complexity (worst case).
- BRP (bounded-error probabilistic polynomial time): decisional problems for which there exists Monte Carlo algorithm of type(3) with polynomial time complexity (worst case).
- ZPP (zero-error probabilistic polynomial time): decisional problems for which there exists Las Vegas algorithm with expected polynomial time complexity (worst case).

$ZPP = RP \cap co-RP$ .

## Poglavje 4

### Chernoff bounds

**Izrek 4.0.1.** Let  $X_1, X_2 \dots X_n$  independent random variables with image  $\{0, 1\}$ .

Let  $p_i = P_r(X_i = x_i)$ ,  $X = \sum_{i=1}^n X_i$  and  $\mu = E(X) = p_1 + \dots + p_n$ .

For every  $\delta \in (0, 1)$ :

$$\begin{aligned} P_r(X - \mu \geq \delta\mu) &\leq e^{-\frac{\delta^2\mu}{3}} \\ P_r(\mu - X \leq \delta\mu) &\leq e^{-\frac{\delta^2\mu}{2}} \\ \implies P_r(|X - \mu| \geq \delta\mu) &\leq e^{-\frac{\delta^2\mu}{3}}. \end{aligned}$$

Probability falls extremely quickly after  $E(X)$ .

**Dokaz 4.0.2.**

$$\begin{aligned}
P_r(X - \mu \geq \delta\mu) &= P_r(X \geq \mu(1 + \delta)) \\
&\stackrel{t \geq 0}{=} P_r(tX \geq t\mu(1 + \delta)) \\
&\stackrel{e^y \geq 0}{=} P_r(e^{tX} \geq e^{t\mu(1 + \delta)}) \\
&\stackrel{\text{Markov}}{\leq} \frac{E(e^{tX})}{e^{t\mu(1 + \delta)}} \\
&\stackrel{4.1}{\leq} \frac{e^{(e^t - 1)\mu}}{e^{t\mu(1 + \delta)}} \\
&\stackrel{4.3}{\leq} e^{-\mu \frac{\delta^2}{3}}.
\end{aligned}$$

$$\begin{aligned}
E(e^{tX}) &= E(e^{tX_1 + \dots + tX_n}) \\
&= E(e^{tX_1} \dots e^{tX_n}) \\
&\stackrel{\text{independent}}{=} \prod_{i=1}^n E(e^{tX_i}) \\
&\stackrel{4.2}{\leq} \prod_{i=1}^n e^{p_i(e^t - 1)} \\
&= e^{(e^t - 1) \sum_{i=1}^n p_i} \\
&= e^{(e^t - 1)\mu}. \tag{4.1}
\end{aligned}$$

$$E(e^{tX_i}) = p_i \cdot e^t + (1 - p_i) \cdot e^0 = 1 + p_i(e^t - 1) \stackrel{1+x \leq e^x}{\leq} e^{p_i(e^t - 1)}. \tag{4.2}$$

Want:

$$e^t - 1 - t(1 + \delta) \leq -\frac{\delta^2}{3} \quad \forall \delta \in (0,1) \tag{4.3}$$

$$t = \ln(1 + \delta)$$

$$f(\delta) = 1 + \delta - 1 - (1 + \delta) \ln(1 + \delta) + \frac{\delta^2}{3} \stackrel{?}{\leq} 0$$

$$f(0) = 0$$

$$f'(\delta) = 1 - \ln(1 + \delta) - 1 + \frac{2}{3}\delta = \frac{2}{3}\delta - \ln(1 + \delta) \stackrel{?}{\leq} 0$$

$$\frac{2}{3}\delta \leq \ln(1 + \delta)$$

$$\delta = 1 : \frac{2}{3} \stackrel{?}{\leq} \ln(2) \approx 0.69 \checkmark$$

$$\begin{aligned}
P_r(\mu - X \leq \delta\mu) &= P_r(X \geq \mu(1 - \delta)) \\
&\stackrel{t \geq 0}{=} P_r(tX \geq t\mu(1 - \delta)) \\
&\stackrel{e^y \geq 0}{=} P_r(e^{tX} \geq e^{t\mu(1 - \delta)}) \\
&\leq \dots \leq \frac{e^{(e^t - 1)\mu}}{e^{t\mu(1 - \delta)}}.
\end{aligned}$$

Want:  $e^t - 1 - t(1 - \delta) \leq -\frac{\delta^2}{2} \forall \delta \in (0, 1)$ :

$$\begin{aligned}
t &= \ln(1 - \delta) \\
f(\delta) &= 1 - \delta - 1 - (1 - \delta) \ln(1 - \delta) + \frac{\delta^2}{2} \stackrel{?}{\leq} 0 \\
f(0) &= 0 \\
f'(\delta) &= -1 + 1 - \ln(1 - \delta) + \delta \stackrel{?}{\leq} 0 \\
\frac{2}{3}\delta &\leq \ln(1 + \delta) \\
\ln(1 - \delta) &\stackrel{?}{\leq} -\delta \checkmark
\end{aligned}$$

■

$$\begin{aligned}
X_i &\sim \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \\
X &= \sum_{i=1}^n X_i \\
\mu &= \frac{n}{2}
\end{aligned}$$

$$\begin{aligned}
P_r(|X - \mu| \geq \sqrt{\frac{3}{2}n \ln(n)}) &= P_r(|X - \mu| \geq \frac{n}{2} \sqrt{\frac{6}{n} \ln(n)}) \\
\mu &= \frac{n}{2}, \delta = \sqrt{\frac{6}{n} \ln(n)}, \\
&\text{for „big“ } n\delta \in (0, 1) \\
&\stackrel{\text{Chernoff}}{\leq} 2e^{-\frac{\frac{n}{2} \frac{6}{n} \ln(n)}{3}} = \frac{2}{n}.
\end{aligned}$$

$$d = \sqrt{\frac{3}{2}n \ln(n)}$$

$$\implies P_r(X \in (\mu - \sqrt{\frac{3}{2}n \ln(n)}, \mu + \sqrt{\frac{3}{2}n \ln(n)})) \geq 1 - \frac{2}{n}.$$



**Trditev 4.0.3.**

Let  $X_1, X_2 \dots$  independent random variables with image  $\{0,1\}$ .

$$P_r(X_i = 1) = \frac{1}{2} \quad \forall i.$$

Let  $X = \sum_{i=1}^{cm} X_i$  where  $c \geq 4$ .

Then  $P_r(X \leq m) \leq e^{-\frac{cm}{16}}$ .

**Dokaz 4.0.4.**

$$\begin{aligned} P_r(X \leq m) &= P_r\left(\frac{cm}{2} - X \geq \frac{cm}{2} - m\right) \\ &= P_r\left(\frac{cm}{2} - X \geq \frac{cm}{2}\left(1 - \frac{2}{c}\right)\right) \\ &\stackrel{\text{Chernoff}}{\leq} e^{-\frac{\frac{cm}{2}\left(1 - \frac{2}{c}\right)^2}{2}} \\ &\quad 1 - \frac{2}{c} \geq \frac{1}{2} \text{ if } c \geq 4 \\ &\leq e^{-\frac{\frac{cm}{2} \cdot \frac{1}{4}}{2}} = e^{-\frac{cm}{16}}. \end{aligned}$$

■

Back to Quicksort.

**Izrek 4.0.5.**

With probability  $\geq 1 - \frac{1}{n}$  Quicksort uses at most  $48n \ln(n)$  comparisons.

**Dokaz 4.0.6.**

For  $s \in S$  define  $S_1^S \dots S_{t_s}^S \neq \emptyset$  sets that include  $s$ ,  $t_s$  - number of comparisons with  $s$  where  $s$  is not a pivot  $+1$ .

Define: iteration  $i$  is successful if  $|S_{i+1}| \leq \frac{3}{4}|S_i|$  ( $\frac{1}{2}$  is too strict).

$$X_i = \begin{cases} 1 & \text{if iteration } i \text{ is successful} \\ 0 & \text{else} \end{cases}$$

$$P_r(X_i = 1) \geq \frac{1}{2}$$

$$S_i : n \rightarrow \frac{3}{4}n \rightarrow \left(\frac{3}{4}\right)^2 n \rightarrow \dots \rightarrow 1.$$

Notice: max number of iteration is  $\log_{\frac{4}{3}}(n) = \frac{\ln(n)}{\ln(4) - \ln(3)}$ .

Probability that we haven't succeeded in  $\log_{\frac{4}{3}}(n)$  steps:

$$P_r\left(\sum_{i=1}^{c \log_{\frac{4}{3}}(n)} X_i < \log_{\frac{4}{3}}(n)\right) \leq P_r\left(\sum_{i=1}^{c \log_{\frac{4}{3}}(n)} Y_i < \log_{\frac{4}{3}}(n)\right) \quad (4.4)$$

$$\stackrel{\text{Chernoff}}{<} e^{-\frac{c \log_{\frac{4}{3}}(n)}{24}} \quad (4.5)$$

$$= e^{-\frac{c \ln(n) \log_{\frac{4}{3}}(e)}{24}} \quad (4.6)$$

$$= \frac{1}{n} \frac{c \log_{\frac{4}{3}}(e)}{24} \quad (4.7)$$

$$\log_{\frac{4}{3}}(e) \approx 3.4, \quad c = 14 \quad (4.8)$$

$$\leq \left(\frac{1}{n}\right)^2 \quad (4.9)$$

4.4 because  $X_i$  not independent,  $Y_i \sim \begin{pmatrix} 0 & 1 \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix}$  independent.

$P_r(t_s \geq c \log_{\frac{4}{3}}(n)) \geq \left(\frac{1}{n}\right)^2$  for one  $s$ .

$c = 14 \implies$  at least  $48 \ln(n)$  iterations with probability  $\leq \left(\frac{1}{n}\right)^2$ .

With probability as least  $1 - \frac{1}{n}$  for all  $s \in S$  it holds that  $s$  has  $\leq 48 \ln(n)$  comparisons with a pivot.

$\implies$  total number of comparisons  $n \cdot 48 \ln(n)$  with probability as least  $1 - \frac{1}{n}$ . ■

## Poglavje 5

# Monte Carlo methods

### 5.1 Example 1

Area of circle =  $\frac{\pi}{4}$ .

$$X_i = \begin{cases} 1 & \text{if you hit the area of circle} \\ 0 & \text{else} \end{cases}$$

$$P_r(X_i = 1) = \frac{\frac{\pi}{4}}{1} = \frac{\pi}{4}.$$

$$E(X_i) = \frac{\pi}{4}.$$

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

$$E(X) = \frac{n \cdot E(X_i)}{n} = E(X_i).$$

### 5.2 Example 2

$I = \int_{\Omega} f(x) dx$  - volume.

$$X_i = \begin{cases} 1 & F(x_i, y_i) \leq z_i \\ 0 & \text{otherwise} \end{cases}$$

$$v \cdot E\left(\frac{\sum_{i=1}^n X_i}{n}\right) = I.$$

### 5.3 $(\epsilon, \delta)$ -approximation

**Definicija 5.3.1**  $((\epsilon, \delta)$ -approximation). A random algorithm gives a  $(\epsilon, \delta)$ -approximation for value  $v$  if the output  $X$  satisfies:

$$P_r(|X - v| \leq \epsilon v) \geq 1 - \delta.$$

**Izrek 5.3.2.** Let  $X_1 \dots X_n$  be independent and identically distributed indicator variables. Let  $\mu = E(X_i)$ ,  $Y = \frac{\sum_{i=1}^m X_i}{m}$ . If  $m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \mu}$ , then  $P_r(|Y - \mu| \geq \epsilon \mu) \leq \delta \implies Y$  is  $(\epsilon, \delta)$ -approximation for  $\mu$ .

**Dokaz 5.3.3.**

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

$$E(X) = mE(x_i) = m\mu$$

$$m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \mu}$$

$$\begin{aligned} P_r(|Y - \mu| \geq \epsilon \mu) &= P_r\left(\left|\frac{X}{m} - \mu\right| \geq \epsilon \mu\right) \\ &= P_r\left(\frac{1}{m} |X - E(X)| \geq \frac{1}{m} \epsilon E(x)\right) \\ &\stackrel{\text{Chernoff}}{\leq} 2e^{-\frac{\epsilon^2 E(x)}{3}} \\ &= 2e^{-\frac{\epsilon^2 \mu m}{3}} \\ &\leq 2e^{-\frac{\epsilon^2 \mu}{3} \cdot \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 \mu}} = \delta. \end{aligned}$$

Back to example 1:

$$E(Y) = \frac{\pi}{4}, \delta = \frac{1}{1000} \text{ (99.9\% sure)}, \epsilon = \frac{1}{10000}$$

$$\implies M = \frac{3 \ln\left(\frac{2}{\frac{1}{1000}}\right)^4}{\pi \left(\frac{1}{10000}\right)^2} \approx 29106.$$

Problems for MC (Monte-Carlo):

- rare events, e.g.  $X \sim \begin{pmatrix} 0 & 10^{100} \\ 1 - 10^{-20} & 10^{-20} \end{pmatrix}$ ,  $E(X) = 10^{80}$

## 5.4 DNF counting

CNF:  $(X_{i_1} \vee \overline{X_{i_2}} \vee X_{i_4}) \wedge (X_{i_1} \vee \overline{X_{i_3}}) \wedge \dots$

DNF:  $(\overline{X_{i_1}} \wedge X_{i_2} \vee \overline{X_{i_4}}) \vee \dots$  - easy to determine if solution exists.

Question: number of solutions to a given DNF?

Observation: CNF  $F$  has a solution  $\iff$  DNF  $\neg F$  has less than  $2^n$  solutions,  
 $n$  is number of samples.

ALG\_1(F):

$x = 0$

for  $i$  in range(1,m+1):

$x_1 \dots x_n$  uniformly random from  $\{0,1\}^n$

if  $F(x_1 \dots x_n) = 1$ :

$x += 1$

return  $\frac{x}{m} \cdot 2^n$

$$Y = \frac{\sum_{i=1}^m X_i}{m}$$

$(\epsilon, \delta)$ -approximation for  $Y$

$$E(Y) = \frac{\text{number of solutions of } F}{2^n} = \frac{c(F)}{2^n}$$

$$m \geq \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2 E(X)} = \frac{3 \ln(\frac{2}{\delta})}{\epsilon^2} \cdot \frac{2^n}{x(F)}$$

$c(F)$  very small  $\rightarrow m$  exponentially big  $\rightarrow$  not good (we need a lot of samples).

### Definicija 5.4.1.

$SC_i = \{(a_1 \dots a_n) \in \{0,1\}^n \text{ such that } F = F_1 \vee \dots \vee F_t, F_i(a_1 \dots a_n) = 1\}$ .

$|SC_i| = 2^{n-l_i}$ ,  $l_i$ : number of values in  $F_i$

$U = \{(i, a) \mid i \in \{1, 2 \dots t\}, a \in SC_i\}$

$U = \sum_{i=1}^t |SC_i| - O(tn)$  (space smaller than  $\{0,1\}^n$ )

$S = \{(i, a) \in U \mid a \in SC_i, a \notin SC_j \ 1 \leq j < i\}$

$|S| = |SC_1| + \dots + |SC_t| = c(F)$ .

ALG\_2(F):

$x = 0$

for  $i$  in range(1,m+1):

```

    (i, a) uniformly random from U (**)
    if (i, a) ∈ S: (*)
        x += 1
    return  $\frac{x}{m} \cdot |U|$ 

```

(\*)  $a \in SC_i \rightarrow O(n)$ ,  $a \notin SC_j \ j = 1 \dots i - 1 \rightarrow O(tn) \implies O(tn), m$  times.

(\*\*): watch for details on how to, e.g.  $x_2, x_2 \wedge x_3$ :  $x_2$  is more probable than  $x_2 \wedge x_3 \rightarrow O(1)$ .

**Izrek 5.4.2.** For  $m = \lceil \frac{3t \ln(\frac{2}{\delta})}{\epsilon^2} \rceil$  algorithm returns  $(\epsilon, \delta)$ -approximation in  $O\left(\frac{t^n n \ln(\frac{2}{\delta})}{\epsilon^2}\right)$  time.

**Dokaz 5.4.3.**  $O(t \cdot n \cdot m)$ .

Insert  $m = \dots$

Prove

$$P_r(Y|U| - c(F) > \epsilon c(F)) < \delta :$$

$$c(F) = |S|, E(Y) = \frac{|S|}{|U|}$$

$$P_r(Y|U| - c(F) > \epsilon c(F)) = P_r(|U|(Y - E(Y)) > \epsilon |U|E(Y)) \leq \delta$$

if

$$m \geq \frac{3 \ln\left(\frac{2}{\delta}\right)}{\epsilon^2 E(Y)} \geq \frac{3 \ln\left(\frac{2}{\delta}\right) t}{\epsilon^2}$$

where

$$E(Y) = \frac{|S|}{|U|} \geq \frac{1}{t}$$

(= if disjoint).

In new space  $E(Y)$  much larger  $\implies m$  smaller.

# Poglavje 6

## Polynomials

Let  $\mathbb{F}$  be a field.

$\mathbb{F}$  can be  $\mathbb{R}, \mathbb{C}, \mathbb{Z}_p, \mathbb{F}_{p^n}$ .

$\mathbb{F}[x_1 \dots x_n]$  algebra of polynomials with values  $x_1 \dots x_n$ .

$f \in \mathbb{F}[x_1 \dots x_n]$

$\deg(f[x_1 \dots x_n]) := \deg(f[x \dots x])$ .

**Izrek 6.0.1.** Let  $p(x_1 \dots x_n) \in \mathbb{F}[x_1 \dots x_n]$  have the degree  $d \geq 0$  and  $p \neq 0$ .

Let  $s \subset \mathbb{F}$  be finite. If  $(r_1 \dots r_n)$  is uniformly at random element from  $S^n$ .

Then  $P_r(p(r_1 \dots r_n) = 0) \leq \frac{d}{|S|}$ .

**Dokaz 6.0.2.** Induction on  $n$ .

$n = 1$ :

$$p(x) = (x - z_1)(x - z_2) \dots (x - z_j)q(z)$$

number of zeros  $\leq$  degree - fact

$$P_r(p(r_1) = 0) = \frac{\text{number of zeros}}{|S|} \leq \frac{d}{|S|}.$$

$n - 1 \rightarrow n$ :

rewrite  $p$  :

$$p(x_1 \dots x_n) = \sum_{i=0}^j x^i p_i(x_2 \dots x_n)$$

$$j \leq d$$

$$\begin{aligned} P_r(p(r_1 \dots r_n) = 0) &= P_r(p(r_1 \dots r_n = 0) \mid p_j(r_2 \dots r_n) = 0) \cdot P_r(p_j(r_2 \dots r_n) = 0) \\ &\quad + P_r(p(r_1 \dots r_n = 0) \mid p_j(r_2 \dots r_n) \neq 0) \cdot P_r(p_j(r_2 \dots r_n) \neq 0) \\ &\leq 1 \cdot \frac{d-j}{|S|} + \frac{j}{|S|} \cdot 1, \end{aligned}$$

because

$$\begin{aligned} P_r(p(r_1 \dots r_n = 0) \mid p_j(r_2 \dots r_n) \neq 0) &\leq \frac{d-j}{|S|} \\ P_r(p_j(r_2 \dots r_n) \neq 0) &\leq \frac{j}{|S|}. \end{aligned}$$

Problem:

Let  $A, B, C \in \mathbb{F}^{n \times n}$ , is  $A \cdot B = C$ ?

Computing  $A \cdot B$ :

- school-book algorithm:  $O(n^3)$ ,
- Strassen algorithm:  $O(n^{2,807\dots})$ ,
- galactic algorithm:  $O(n^{2,372\dots})$  - has enormous constants.

`RAND_ACB(A,B,C) :`

`for i in range(1,k+1):`

`x uniformly at random from  $\{0,1\}^n$`

`if  $A \cdot (B \cdot x) \neq x$ :`

`return false`

`return true`



$O(kn^2)$ .

If  $A \cdot B = C$ , algorithm returns true.

If  $A \cdot B \neq C$ :

$$\begin{aligned} P_r(ABx = Cx) &= P_r((AB - C)x = 0) \\ &= P_r(\|(AB - C)x\|^2 = 0) \stackrel{\text{Poly}}{\leq} \frac{2}{3}. \end{aligned}$$

$\|(AB - C)x\|^2$  - polynomial in  $x_1 \dots x_n$  of degree 2.

If  $A \cdot B \neq C$ , then algorithm return false with probability at least  $1 - \left(\frac{2}{3}\right)^k$ .

Problem:

1-factor in bipartite graphs.

$|V(g)| = 2n$ .

Represent  $G$  with  $n \times n$  matrix  $Z = (Z_{ij})_{i,j=1}^n$

$$Z_{ij} = \begin{cases} X_{ij} & \text{if } a_i b_j \in E(x) \\ 0 & \text{else} \end{cases} \quad (X: \text{variable})$$

$$\begin{aligned} \det Z(x_{11} \dots x_{nn}) &= \sum_{\pi \in S_n} \text{sign}(\pi) z_{1,\pi(1)} \dots z_{n,\pi(n)} \\ &= \sum_{\pi \in S_n, \pi \text{ defines 1-factor}} \text{sign}(\pi) x_{1,\pi(1)} \dots x_{n,\pi(n)}. \end{aligned}$$

$\det Z \neq 0 \iff G$  has 1-factor.

```

Rand_1factor(G):
  construct Z with variables x11 ... xnn
  for i in range(1,k+1):
    u <- uniformly at random from 1,2..2n-1n2 (r11 ... rnn)
    compute d = det Z(r11 ... rnn)
    if d != 0:
      return true
  return false

```

Complexity:  $k \cdot$  computing determinant:  $O(n^3)$  (Gaussian elimination).  
or apply approximation algorithm:

- if  $G$  has no 1-factor it always returns false,
- if  $G$  has 1-factor, it returns true with probability at least  $1 - \left(\frac{n}{2n}\right)^k = 1 - \left(\frac{1}{2}\right)^k$  ( $k$  konstant, larger set  $\implies$  smaller  $k$  needed).

# Poglavje 7

## Random graphs

### 7.1 $G(n,p)$ model

$G$  is a random Erdős-Rényi graph if it has  $n$  vertices and each pair of vertices is connected with probability  $p$ .

*Primer.*  $G\left(5, \frac{1}{2}\right)$ .

$E(\text{edges in } G \text{ from } G(n,p)) = \sum_{1 \leq i < j \leq n} E(X_{ij}) = \binom{n}{2}p$ .

$$X_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ have edge} \\ 0 & \text{otherwise} \end{cases}$$

$p$  can be function of  $n$ .

$Y_v$  : degree of  $v$ .

$$E(Y_v) = (n-1)p.$$

#### **Definicija 7.1.1.**

We say that a random graph has some property almost surely (A.S.) if  $P_r(G \in G(n,p) \text{ has property}) \xrightarrow{n \rightarrow \infty} 1$ .

#### **Trditev 7.1.2.**

Let  $p$  be constant. Then  $G \in G(n,p)$  has diameter 2 A.S.

#### **Dokaz 7.1.3.**

Let  $u, v \in V(G)$   
 $X_w = \begin{cases} 1 & \text{if } uw \in E(G) \text{ in } vw \in E(G) \\ 0 & \text{otherwise} \end{cases}$   
 $P_r(X_w = 1) = p^2$   
 $P_r(X_w = 0 \text{ for all } w \neq u, v) = (1 - p^2)^{n-2}.$   
 $P_r(G \text{ has diameter } > 2)$   
 $= P_r(X_w = 0 \text{ for all } w \notin u, v \text{ for some } u, v)$   
 $\leq \binom{n}{2} (1 - p^2)^{n-2} \xrightarrow{n \rightarrow \infty} 0;$   
 $\binom{n}{2}$  - polynomial,  $e^{\dots}$  - exponent.

$p = f(n)$   
 $\frac{1}{n}, \frac{1}{n^3}, \frac{\log n}{n}$

**Izrek 7.1.4.** (without proof)

Let  $p$  be a function of  $n$ : let  $G \in G(n, p)$ :

- $np < 1$  -  $G$  A.S. disconnected with connected components of size  $O(\log n)$
- $np = 1$  -  $G$  A.S. has 1 large component of size  $O\left(n^{\frac{2}{3}}\right)$
- $np = c > 1$  -  $G$  A.S. has giant component of size  $dn$ ,  $d \in (0, 1)$
- $np \leq (1 - \epsilon) \ln n$  -  $G$  A.S. disconnected with isolated vertices
- $np > (1 - \epsilon) \ln n$  -  $G$  A.S. connected.

**Izrek 7.1.5.**

Let  $np = \omega(n) \ln(n)$  for  $\omega(n) \rightarrow \infty$  „very slowly“ think of  $\omega(n) = \log(\log n)$ , then  $\text{diam}(G)$  in  $\Theta\left(\frac{\ln n}{\ln(np)}\right)$  for  $G$  in  $G(n, p)$ .

**Lema 7.1.6.**

Let  $S \subset V(G)$ ,  $|S| = cn$  for  $c \in (0, 1]$  and  $v \notin S$ .

then  $cnp(1 - \omega^{-\frac{1}{3}}) \leq N_S(v) \leq cnp(1 + \omega^{-\frac{1}{3}})$  A.S. ( $\omega^{-\frac{1}{3}} \rightarrow 0$  very slowly).

**Dokaz 7.1.7.** (Lemma):

$E(N_S(v)) = c \cdot n \cdot p$ ,  $\delta = \omega^{-\frac{1}{3}}$

$$\begin{aligned}
P_r(|N_s(v) - cnp| \geq \delta cnp) &\stackrel{\text{Chernoff}}{\leq} 2e^{-\frac{\omega^{-\frac{2}{3}} cnp}{3}} \\
&= 2e^{-\frac{cnp}{3\omega(n)^{\frac{2}{3}}}} \xrightarrow{n \rightarrow \infty} 0.
\end{aligned}$$

For all  $v$ :  $n \cdot 2e^{-\frac{cnp}{3\omega(n)^{\frac{2}{3}}}} \xrightarrow{n \rightarrow \infty} 0$ .

**Dokaz 7.1.8.** (Theorem):

$k$  be such that  $\sum_{i=0}^{k-1} |N_i| \leq \frac{n}{2}$ ,  $\sum_{i=0}^k |N_i| > \frac{n}{2}$ .

$$|N_0| = 1$$

$$|N_i| \leq |N_{i-1}| \cdot n \cdot p \cdot (1 + \omega^{-\frac{1}{3}}):$$

$$|S| \leq n, \quad np(1 + \omega^{-\frac{1}{3}})\text{-each element.}$$

$$\begin{aligned}
k &= \frac{\log\left(\frac{n}{3}\right)}{\log\left(n \cdot p \cdot \left(1 + \omega^{-\frac{1}{3}}\right)\right)} \\
&= \log_{np(1 + \omega^{-\frac{1}{3}})} \frac{n}{3} = \Theta\left(\frac{\ln(n)}{\ln(np)}\right). \\
|N_{\leq k}| &= |N_1 \cup \dots \cup N_k|.
\end{aligned}$$

$$\begin{aligned}
|N_{\leq k}| &\leq \sum_{i=0}^k (np(1 + \omega^{-\frac{1}{3}}))^i \\
&= \frac{(np(1 + \omega^{-\frac{1}{3}}))^{k+1} - 1}{np(1 + \omega^{-\frac{1}{3}}) - 1} \\
&< \frac{np(1 + \omega^{-\frac{1}{3}})^{k+1}}{\frac{1}{2}np(1 + \omega^{-\frac{1}{3}})} \\
&= 2np(1 + \omega^{-\frac{1}{3}})^k \\
&\stackrel{k}{=} 2 \cdot \frac{n}{3} \text{ haven't covered all} \\
&\implies \text{diam}(G) > k \text{ bound from below.}
\end{aligned}$$

$$N_i \subseteq S$$

$$\frac{1}{2}np(1 - \omega^{-\frac{1}{3}}) \cdot |N_{i-1}| \leq |N_i|$$

$$\begin{aligned}
n &\geq \sum_{i=0}^k |N_i| \\
&\geq \sum_{i=0}^k \left( \frac{1}{2} np \left( 1 - \omega^{-\frac{1}{3}} \right) \right)^i \\
&= \frac{\left( \frac{1}{2} np \left( 1 - \omega^{-\frac{1}{3}} \right) \right)^{k+1} - 1}{\frac{1}{2} np \left( 1 - \omega^{-\frac{1}{3}} \right) - 1} \\
&\geq \left( \frac{1}{2} np \left( 1 - \omega^{-\frac{1}{3}} \right) \right)^k / \ln
\end{aligned}$$

$$\frac{\ln n}{\ln(np)} \approx \frac{\ln n}{\ln\left(\frac{1}{2} np \left( 1 - \omega^{-\frac{1}{3}} \right)\right)} \geq k.$$

$$\implies w \in S'.$$

Number of neighbors in  $N_k$  A.S.  $\geq 1$ ,

$$|N_k| \geq \left( \frac{1}{2} np \left( 1 - \omega^{-\frac{1}{3}} \right) \right)^k \approx c \cdot n$$

$$\implies \text{diam}(G) = k + 1 \text{ A.S.}$$

### 7.1.1 Scale free property

$$G \in G(n, p).$$

In real world:  $p(k)$  = proportion of degree  $k$  vertices.

$$\log(p(k)) = -\gamma \cdot \log k$$

$$p(k) = k^{-\gamma}.$$

Internet:  $\gamma \approx 3.42$ ,

protein reactions:  $\gamma \approx 2.89$ .

## 7.2 Barbási-Albert Model

B.A. model.

Start with  $m$  nodes.

Grow:

- add node  $v$ ,

- add  $m$  edges from  $v$  (to  $u$ ),
- for each new edge:  $P(v \sim u) = \frac{deg_u}{\sum_x deg_x}$ .

**Izrek 7.2.1.**

B.A. model has scale free property, in particular

$$p_k = \frac{2m(m+1)}{k(k+1)(k+2)}.$$

**Definicija 7.2.2.**

$p_n(k)$ : expected proportion of degree  $k$  vertices in graph with  $k$  vertices,

$$p_k := \lim_{n \rightarrow \infty} p_n(k).$$

**Dokaz 7.2.3.**

$p_n(k) \cdot n$ : expected number of degree  $k$  vertices,

$p_n(k)n \cdot \sum_u \frac{k}{deg_u} m = p_n(k) \cdot \frac{k}{2}$ : expected number of degree  $k$  vertices changing into degree  $k+1$  vertices.

$$\sum_u deg_u = 2|E|$$

$$p_{n+1}(k) \cdot (n+1) = p_n(k) \cdot n - p_n(k) \cdot \frac{k}{2} + p_n(k-1) \cdot \frac{k-1}{2}, \text{ where}$$

$$p_n(k) \cdot n: \text{ degree } k \rightarrow k,$$

$$p_n(k) \cdot \frac{k}{2} : k \rightarrow k+1,$$

$$p_n(k-1) \cdot \frac{k-1}{2} : k-1 \rightarrow k.$$

For  $n$  very big (very close to limit):

$$p_n \cdot (n+1) = p_k \cdot n - p_{k-1} \cdot \frac{k}{2} + p_{k-1} \cdot \frac{k-1}{2}$$

$$\implies p_k = \frac{k-1}{k+2} p_{k-1}.$$

For degree  $m$ :

$$(n+1) \cdot p_{n+1}(m) = p_n(m) \cdot n - p_n(m) \cdot \frac{m}{2} + 1$$

$$\begin{aligned} p_m &= \frac{2}{m+2} \\ \implies p_{m+1} &= \frac{2}{m+2} \cdot \frac{m}{m+3} \\ \implies p_{m+2} &= \frac{2m(m+1)}{(m+2)(m+3)} \\ \implies p_k &= \frac{2m(m+1)}{k(k+1)(k+2)}. \end{aligned}$$

# Poglavje 8

## Markov chains

$\Omega$ : finite set (of states).

**Definicija 8.0.1** (Markov chain).

(Discrete time) Markov chain is a sequence of random variables  $X = X_0, X_1, X_2 \dots$  with image  $\Omega$  and properties:

- $P(X_{i+1} = x \mid X_i = x_i, X_{i-1} = x_{i-1} \dots X_0 = x_0) = P(X_{i+1} = x \mid X_i = x_i),$
- $PX_{i+1} = x \mid X_i = y = P(X_1 = x \mid X_0 = y)$  - time is homogenous.

*Primer.*

$$\Omega = \mathbb{Z}_5$$

$$P(X_{i+1} = x + 1 \mid X_i = x) = \frac{1}{2}$$

$$P(X_{i+1} = x - 1 \mid X_i = x) = \frac{1}{2}.$$

**Definicija 8.0.2** (Transition matrix).

$$\Omega = \{x_1 \dots x_n\}$$

$$p_{ij} = P(X_{t+1} = j \mid X_t = i)$$



$$\begin{bmatrix} p_{11} & \dots & \\ p_{1n} & & \\ \vdots & & \vdots \\ p_{n1} & \dots & p_{nn} \end{bmatrix}.$$

**Definicija 8.0.3** (Transition graph).

Edge between states  $i$  and  $j$  exists if  $p_{ij} > 0$ .

$P$  is stochastic matrix:

$$p_{ij} \in [0,1]$$

$$\sum_j p_{ij} = 1.$$

We choose beginning state randomly.

$$q(0) = (q_1(0) \dots q_n(0))$$

$$P(X_0 = i) = q_i(0).$$

$$\text{Let } q(t) = (q_1(t) \dots q_n(t))$$

$$P(X_t = i) = q_i(t).$$

$$\text{It holds: } q(t) = q(t-1) \cdot P = q(0) \cdot P^t.$$

$$\begin{bmatrix} 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}$$

$$q(0) = (1, 0, 0, 0, 0)$$

$$q(1) = (1, \frac{1}{2}, 0, 0, \frac{1}{2})$$

$$q(2) = (\frac{1}{2}, 0, \frac{1}{4}, \frac{1}{4}, 0)$$

$$\vdots$$

**Definicija 8.0.4.**

- Distribution  $\pi$  is stationary if  $\pi = \pi \cdot P$ ,
- $f_{ij}$ : probability that  $X_t = x_j$  for some  $t$  assuming  $X_0 = x_i$ ,

- $h_{ij}$ : expected number of steps needed to get to state  $X_j$  starting in  $X_i$  (hitting time),
- $N(i, t, q(0))$ : expected number of times we visit  $x_i$  after  $t$  steps starting with distribution  $q(0)$ ,
- $\forall f_{ij} > 0 \iff$  transition graph is strongly connected  $\iff$  we say the chain is irreducible,
- M.C. is aperiodic if there is no  $c \in \{2, 3, 4, \dots\}$  such that all lengths of cycles are divisible by  $c$ .

**Izrek 8.0.5.**

Let  $X$  be finite irreducible M.C. Then:

- there exists unique stationary distribution  $\pi = (\pi_1 \dots \pi_n)$ ,
- $f_{ii} = 1, h_{ii} = \frac{1}{\pi_i}$ ,
- $\lim_{t \rightarrow \infty} \frac{N(i, t, q(0))}{t} = \pi_i$  - approaches  $\pi$  regardless of  $q(0)$ ,
- if  $X$  is aperiodic:  $\lim_{t \rightarrow \infty} q(0) \cdot P^t = \pi$ .

*Primer.*

$$P = \begin{bmatrix} 0 & \frac{1}{2} & \dots & \frac{1}{2} \\ \frac{1}{2} & 0 & \dots & 0 \\ \vdots & & & \vdots \\ \dots & & \frac{1}{2} & 0 \end{bmatrix}$$

$$\pi = (\frac{1}{n} \dots \frac{1}{n})$$

$$h_{i,i} = n$$

$$n = h_{i,i} = 1 + \frac{1}{2}h_{i-1,i} + \frac{1}{2}h_{i+1,i}, \quad h_{i-1,i} = h_{i+1,i}$$

$$n - 1 = h_{i-1,i}$$

$$E(\text{steps around}) \leq h_{0,1} + h_{1,2} + \dots + h_{n-1,n} \leq n(n-1).$$

## 8.1 2-SAT

Recall: k-SAT:

$$F = C_1 \wedge \cdots \wedge C_m$$

$$C_i = X_{i1} \vee \cdots \vee X_{ik}.$$

3-SAT: NP complete.

Algorithm:

```
def rand2SAT(F):
    b0 = (b_00 ... b_n0)
    for i in range(t):
        if F(bi) = 1:
            return True
        Cl <- clause that is False
        xj <- uniformly at random from x11 and x12
        bi+1 = (b_0i ... not b_ji ... b_ni)
    if F(Xt) = 1:
        return True
    return False
```

**Izrek 8.1.1.**

If  $k = 8n^2$ , then  $P(\text{rand2SAT} = \text{True} \mid \text{correct answer is True}) \geq \frac{3}{4}$ .

**Dokaz 8.1.2.** Let  $a = (a_1 \dots a_n)$  be a correct solution.

Let  $X_i$  = Hamming distance from  $b^i$  to  $a$ .

Goal: bound  $h_{n,0}$ .

$P(\text{distance of } b^{i+1} \text{ to } a \text{ is } j-1 \mid \text{distance of } b^i \text{ to } a \text{ is } j) \geq \frac{1}{2}$ .

$$P = \begin{bmatrix} 0 & 1 & \dots & 0 \\ \frac{1}{2} & 0 & \dots & 0 \\ \vdots & & & \vdots \\ \dots & & 1 & 0 \end{bmatrix}$$

$$\pi \stackrel{?}{=} \pi P$$

$$\pi = (\frac{1}{2n}, \frac{1}{n} \dots \frac{1}{n}, \frac{1}{2n})$$

By theorem

$$h_{i,i} = \frac{1}{\pi_i} = n \text{ for } i = 1, 2 \dots n-1$$

$$h_{0,0} = h_{n,n} = 2n$$

$$n = h_{i,i} = 1 + \frac{1}{2}h_{i+1,i} + \frac{1}{2}h_{i-1,i}$$

$$h_{i+1,i} \leq 2n$$

$$i = 0 : 2n = h_{0,0} = 1 + h_{1,0} \implies h_{1,0} < 2n$$

$$h_{n,0} \leq h_{n,n-1} + \dots + h_{1,0} \leq 2n^2$$

$$E(\text{steps in algorithm to reach corrc solution}) = E(Z) \leq 2n^2$$

$$P(\text{algorithm hasn't reached correct solution after } 8n^2 \text{ steps})$$

$$= P(Z > 8n^2) \stackrel{\text{Markov}}{\leq} \frac{E(Z)}{8n^2} \leq \frac{1}{4}.$$

## 8.2 Generating a uniformly random element of a set

$\Omega$ : set.

Let  $G$  be a symmetric graph on  $\Omega$ .

We form M.C:

$$P_{x,y} = \begin{cases} \frac{1}{M} & \text{if } x \neq y \wedge x \sim y \\ 0 & \text{if } x \neq y \wedge x \not\sim y \\ 1 - \frac{|N(x)|}{M} & \text{if } x = y \end{cases}$$

$$M \geq \max_{v \in \Omega} |N(v)|.$$

If  $G$  is connected  $\implies$  M.C. is irreducible.

$$\pi = (\frac{1}{|\Omega|} \dots \frac{1}{|\Omega|})$$

$$\pi \stackrel{?}{=} \pi P$$

$$\begin{aligned}
(\pi P)_x &= \sum_y \pi_y P_{y,x} \\
&= \sum_{y \in N(x)} \frac{1}{M} \cdot \frac{1}{|\Omega|} + \frac{1}{|\Omega|} \left( 1 - \frac{|N(x)|}{M} \right) = \frac{1}{|\Omega|} = \pi_x.
\end{aligned}$$

$\implies$  if we walk on the Markov chain long enough, we end up in state  $x$  with probability  $\pi_x = \frac{1}{|\Omega|}$

$\implies$  we can sample uniformly.

*Primer.*

$G$  graph, finding largest independent set ( $\forall u, v : u \not\sim v$ ) is NP-complete.

Lets try sampling a uniformly random independent set

$\Omega = \{\text{independent sets}\}$

$u \sim v$  if  $|u \triangle v| = 1$  ( $(u \cup \{el\}) = v$ )

M.C.:  $X_0 =$  arbitrary independent set

$X_{i+1}$ :

- pick uniformly at random  $v \in V(G)$ ,
- if  $v \in U$  then  $X_{i+1} = U \setminus \{v\}$ ,
- if  $U \cup \{v\}$  is independent then  $X_{i+1} = U \cup \{v\}$ ,
- else  $X_{i+1} = U$ .

$M$  is number of vertices

$\implies \forall u \in \Omega : \lim_{t \rightarrow \infty} P(X_t = u) = \frac{1}{|\Omega|}$ .

Note: irreducible;  $U \rightarrow \emptyset \rightarrow V$ , aperiodic.

## 8.3 Metropolis algorithm

$\Omega$ : set,

$\pi$ : chosen distribution on  $\Omega$ .

Make  $G$  graph on  $\Omega$

$$P_{x,y} = \begin{cases} \frac{1}{M} \cdot \min\left(1, \frac{\pi_y}{\pi_x}\right) & \text{if } x \neq y \wedge x \sim y \\ 0 & \text{if } x \neq y \wedge x \not\sim y \\ 1 - \sum_{y \in N(x)} & \text{if } x = y \end{cases}$$

$$M \geq \max_{v \in \Omega} |N(v)|$$

$$\pi \stackrel{?}{=} \pi P$$

$$\begin{aligned} (\pi P)_x &= \sum_y \pi_y P_{y,x} = \sum_{y \in N(x)} \pi_y \frac{1}{M} \min\left(1, \frac{\pi_y}{\pi_x}\right) + \pi_x \left(1 - \sum_{y \in N(x)} \frac{1}{M} \min\left(1, \frac{\pi_y}{\pi_x}\right)\right) \\ &= \sum_{y \in N(x), \pi_y \geq \pi_x} \pi_y \frac{1}{M} \cdot 1 + \sum_{y \in N(x), \pi_y < \pi_x} \pi_y \frac{1}{M} \frac{\pi_y}{\pi_x} + \pi_x \\ &\quad - \sum_{y \in N(x), \pi_y \geq \pi_x} \pi_x \frac{1}{M} \frac{\pi_y}{\pi_x} - \sum_{y \in N(x), \pi_y < \pi_x} \frac{1}{M} \cdot 1 \\ &= \pi_x. \end{aligned}$$

*Primer.*

$$\Omega = \mathbb{Z} \cap [-1000, 1000]$$

$$\pi \sim e^{-\frac{(x-\mu)^2}{2\delta}}$$

```

X0 arbitrary
for i = in range(1,m):
    y <- uniformly from Xi+1,Xi-1
    M <- uniformly from [0,1]
    if M ≤  $\frac{\pi(y)}{\pi(x)}$ :
        Xi+1 = y
    else:
        Xi+1 = Xi
return Xm

```

*Primer.*

Find maximum of a positive function  $f$ .

Use metropolis algorithm to sample proportional to  $f$ .

Note: all I need to know is ratios  $\frac{f(y)}{f(x)}$ .

Back to independent sets.

$$G = (V, E)$$

$\Omega$  = independent sets.

$$\lambda \in (1, \infty)$$

$$\pi(u) \sim \lambda^{|u|}$$

$$\pi(u) = \frac{\lambda^{|u|}}{\sum_{v \text{ independent set}} \lambda^{|v|}}.$$

How to calculate the sum?

No problem: only need proportions.

$X_0$ : arbitrary independent set.

$X_i \rightarrow X_{i+1}$ :

- we pick  $v \in V$  uniformly at random,
- if  $v \in X_i \implies$ 
  - $X_{i+1} = X_i \setminus \{v\}$  with probability  $\frac{1}{\lambda} = \min\{1, \frac{\pi_v}{\pi_x}\},$
  - $X_{i+1} = X_i$  with probability  $1 - \frac{1}{\lambda},$
- if  $v \in X_j$  and  $X_i \cup \{v\}$  is independent  $\implies X_{i+1} = X_i \cup \{v\},$
- otherwise  $X_{i+1} = X_i.$

*Primer.* Bayes:  $P(A | B) = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}.$

$B \leftarrow$  machine is giving values, e.g.  $y_1 = 0.05, y_2 = -0.1, y_3 = 0.07, y_4 = 3.$

We believe  $B \sim N(\mu, 0.05).$

$\mu = \text{laplacian}(0, 0.01).$

$$P(\mu | B) = \frac{e^{\frac{|\mu|}{0.01}} e^{-\sum \frac{(x_i - \mu)^2}{0.05}}}{\int \dots}$$

Integral is difficult to calculate.

Sample  $\mu$  with Metropolis algorithm.

## 8.4 M.C. for 1-factor in bipartite graphs

$G$  regular graph

$$|A| = |B|.$$

How to find 1-factor?

Augmenting paths.

Let  $M$  be (suboptimal) matching.

If we find  $s - t$  path, we switch edges and get bigger matching.

Starting point.

$G$   $d$ -regular graph.

Graph  $G = (A \cup B, E)$ ,  $M$  suboptimal matching.

- Add  $s$  and add directed edges to vertices in  $A$  that are not matched with weight  $d$ ,
- add  $t$  and add directed edges to vertices in  $B$  that are not matched with weight  $d$ ,
- orient edges in  $M$  from  $B$  to  $A$  that weight  $d - 1$ ,
- orient edges in  $E \setminus M$  from  $B$  to  $A$  that weight  $1$ ,
- we add edge from  $t$  to  $s$  that weight  $(|A| - |M|)d$ .

Observation:

- for each vertex  $x$  :  $\deg^-(x) = \deg^+(x)$  (out weights = in weights),
- if  $|A| > |M|$ , then graph is eulerian  $\implies$  there is an augmenting path.

How to find  $s - t$  path?

Do a random walk.

Expected time to get from  $s$  to  $t$  is  $h_{s,t}$

$$\frac{1}{\pi(s)} = h_{s,s} = h_{s,t} + 1.$$

**Lema 8.4.1.**

Let  $X$  be a M.C. defined as a random walk on directed (weighted) graph with  $\deg^-(x) = \deg^+(x)$  for each  $x$ . Then the stationary distribution is



$$\pi = \left[ \frac{\deg^+(x_i)}{|E|} \right]_{i=1}^n.$$

$w_{ij}$ : weight from  $i$  to  $j$ .

#### Dokaz 8.4.2.

$$\pi P = \pi \left[ \frac{w_{ij}}{\deg^+(x_i)} \right]_{i,j=1}^n = \left[ \frac{\sum_j w_{ji}}{|E|} \right]_{i=1}^n = \left[ \frac{\deg^-(x_i)}{|E|} \right]_{i=1}^n = \left[ \frac{\deg^+(x_i)}{|E|} \right]_{i=1}^n.$$

$$h_{s,s} = \frac{1}{\pi_s} \leq \frac{|E|}{\deg^+(s)} \leq \frac{3(|A|-|M|)d+|M|(d-1)+(|A|-|M|)d+|M|(d-1)}{(|A|-|M|)d} \leq \frac{4|A|}{|A|-|M|}.$$

Expected time to find augmenting path  $\leq \frac{4|A|}{|A|-|M|}$ .

$$|A| = n$$

Expected time to find 1-factor  $\leq \frac{4n}{n-1} = 4n \sum_{i=1}^{n-1} \frac{1}{i} \leq 4n(1 + \ln n)$  - in  $O(n \log n)$ .

### 8.4.1 Network centrality

Degree as measure - natural idea.

Use M.C: walk randomly on the network, those that are visited more often are more important.

Pagerank.

Let  $A$  be the adjacency matrix of  $G$ .

$$P_{ij} = \alpha \frac{A_{ij}}{\deg_i} + (1 - \alpha) \frac{1}{n};$$

$\alpha$ : normal random walk,

$1 - \alpha$ : jump to any.

$$\alpha = 0.85.$$

## Poglavje 9

# Randomized incremented constructions (RIC)

Observation:

Let  $S$  be a set of  $n$  distinct elements.

Let  $X_1 \dots X_n$  be a random permutation of the elements.

Let  $S_i = \{X_1 \dots X_i\}$ .

$$P(X_i = \min(S_i)) = \frac{1}{i}.$$

$$Y = |\{j \in \{1 \dots n\} \mid j = \text{minimal of } S_j\}|$$

$$Y = Y_1 + \dots + Y_n$$

$$Y_j = \begin{cases} 1 & \text{if } j = \min S_j \\ 0 & \text{otherwise} \end{cases}$$

$$E(Y) = \sum_{i=1}^n E(Y_i) = \sum_{i=1}^n \frac{1}{i} \text{ in } O(\log n).$$

Alg():

```
X1 .. Xn = random permutation of S
```

```
min = X1
```

```
for i in range(1,n+1):
```

```
    if Xi < min:
```

```
        print("HA")
```

```
    min = Xi
```

We get  $O(\log n)$  „HA“ printed.

Incremental construction (IC).

Input  $S = \{s_1 \dots s_n\}$ .

We will build structures  $DS(S_i)$ :

$DS(S_1 \rightarrow \dots \rightarrow DS(S_n))$ .

$DS(S_n)$  will help us give answer.

Randomized: permute  $S$  at the beginning.

## 9.1 Quicksort as RIC

$S$ : set of elements we want to order.

$X_1 \dots X_n$ : random permutation of  $S$ .

$S_i = \{X_1 \dots X_i\}$ .

$S_i$  splits  $\mathbb{R}$ .

Define  $DS(S_i)$ :

- save intervals: each interval will be saved by endpoints,
- for each interval we will be saving its points,
- for each  $X_j$ ,  $j > i$  we will save in which interval it is,
- for each left point of the interval we will save the right point.

QuicksortRIC(S):

```
# start of DS(Si)
I=[(-∞,∞)]
P[(-∞,∞)] = S
for each Xi:
    Int(Xi) = (-∞,∞)
Next(∞) = ∞
# end of DS(Si)
for i in range(1,n+1):
    Ii = Int(Xi) = (Xj,Xk)
```

```

    Ii1 = (Xj, Xi)
    Ii2 = (Xi, Xk)
    for Xl ≠ Xi, Xi ∈ P(I):
        add Xl to P(Ii1) or P(Ii2) depending on Xl < Xi or Xl > Xi
    Next(Xj) = Xi
    Next(Xi) = Xk
    return [Next(−∞), Next(Next(−∞)) ..]

```

Similarity to quicksort: splitting intervals.

Analysis:

for set  $i$ , we need  $O(|P(I_i)|)$ ,

$E(|P(I_i)|) = ?$

e.g.

if  $x_4 = a_4$ :

if  $x_4 = a_2$ :

$P(X_i = a_j) = \frac{1}{i} \quad j \in \{1, 2 \dots i\}$ .

Expected value of steps in iteration  $i$

$$\sum_{j=1}^i \frac{1}{i} (P((a_{j-1}, a_j)) + P((a_j, a_{j+1}))) \leq \frac{1}{i} 2(n-i) \leq \frac{2n}{i}$$

$$\begin{aligned}
 E(\text{number of steps in QuicksortRIC}) &\leq \sum_{i=1}^n \frac{2n}{i} \\
 &\leq 2n(1 + \log n) \quad \rightarrow \text{ in } O(n \log n).
 \end{aligned}$$

## 9.2 Linear programming

Task: maximize  $f(x_1 \dots x_n) = c_1 x_1 + \dots + c_d x_d$ .

Constraints:

$$a_{11}x_1 + \dots + a_{1d}x_d \leq b_1$$

$$\vdots$$

$$a_{n1}x_1 + \dots + a_{nd}x_d \leq b_n.$$

Geometric interpretation.

Cases:

- infeasible region
- unbounded
- .

Alg:

- simplex algorithm worst case  $O(2^n)$ ,
- interior point method (polynomial algorithm).

Seidel's algorithm:

running in expected  $O(n)$  time when  $d$  is constant.

One dimension.

$$\begin{aligned} \max \quad & cx \\ & a_1x \leq b_1 \\ & \vdots \\ & a_nx \leq b_n, \end{aligned}$$

where  $n$  is number of constraints.

- $a_i$  positive:  $(-\infty, \frac{b_i}{a_i}]$ ,
- $a_i$  negative:  $[\frac{b_i}{a_i}, \infty)$ .

$a_i \neq 0$ .

Alg:

$$R = \min_i \left\{ \frac{b_i}{a_i}; a_i > 0 \right\},$$

$$L = \max_i \left\{ \frac{b_i}{a_i}; a_i < 0 \right\},$$

if  $L > R$ : program infeasible,

else:

if  $c > 0$ : return  $R$ ,

if  $c < 0$ : return  $L$ .

2-dim: assume general position.

$$\begin{aligned}
 & \max c_1x + c_2y \\
 & a_{11}x + a_{12}y \leq b_1 \\
 & \vdots \\
 & a_{n1}x + a_{n2}y \leq b_n \\
 & x \leq M \text{ or } x \geq -M \\
 & y \leq M \text{ or } y \geq -M.
 \end{aligned}$$

$\leq, \geq$  depending on  $c_1, c_2$ .

Notation:

$h_i$ : halfspace defined by  $a_{i1}x + a_{i2}y \leq b_i$ ,

$m_i$ : added halfspaces,

$l_i$ : line that bounds.

Alg:

- first randomly permute  $h_i$ ,
- $H_i = \{m_1, m_2, h_1 \dots h_i\}$ ,
- $v_i \in \cap H_i$  optimal solution after  $i$  constraints,
- $v_0 = (\pm M, \pm M)$ ,
- inductively add  $h_i$ .

Cases:

if  $v_{i-1} \in h_i \implies v_i = v_{i-1}$ ,

if  $v_{i-1} \notin h_i \implies v_i \in h_i$ :

$$a_{i1}x + a_{i2}y = b_i$$

$$a_{i1} \text{ or } a_{i2} \neq 0, \text{ e.g. } a_{i1};$$

$$x = \frac{b_i - a_{i2}y}{a_{i1}}.$$

Insert  $x$  in all constraints  $\implies$  linear program in 1-dim,  $i$  (i-1?) constraints  
 $\implies$  get  $v_i$  in  $O(i)$ .

Analysis:

- worst case:  $\sum_{i=1}^n O(i) = O(n^2)$ ,
- expected:  $E(X) = \sum_{i=1}^n E(X_i)$ ,
- $X_i$  = running time of  $i$ -th iteration,
- $X_i = \begin{cases} O(1); & \text{case 1} \\ O(i); & \text{case 2} \end{cases}$
- $P(\text{case 2}) \leq \frac{2}{i}$  - optimal point on at most 2 lines,
- $E(X) \leq \sum_{i=1}^n O(1) \cdot 1 + O(i) \cdot \frac{2}{i} = O(n)$ .

$d$ -dim

- constraints define half-spaces,
- boundary is hyperplane ( $d - 1$  dimensional),
- general position: intersection of  $d - i$  hyperplanes is  $i$  dimensional, intersection of  $d + 1$  hyperplanes is  $\emptyset$ .

Alg:

first add  $X_i \leq M$  or  $X_i \geq -M$  depending on  $c_i$ ,

random permutation  $(h_1 \dots h_n)$ ,

$$H_i = \{m_1 \dots m_d, h_1 \dots h_i\},$$

$$v_0 \in \cap \partial m_i,$$

inductively add  $h_i$ :

$$v_{i-1} \in h_i \implies v_i = v_{i-1},$$

$v_{i-1} \notin h_i \implies$  we need to solve LP in  $d - 1$  dimensions with  $i$  constraints ( $O(i)$  expected),

$$P(v_{i-1} \notin h_i) \leq \frac{d}{i},$$

$$E(X) \leq \sum_{i=1}^n O(1) + \frac{d}{i} O(i) = O(n).$$

$X$ : running time.

Careful implementation runs in  $O(d!n) \implies$  very useful for low dimensions.

Problem: let  $P$  be convex polygon given by ordered set of vertices

$$y = a_i x + b_i.$$

Find largest disc embeddable in  $P$ .

Input:  $P_1 \dots P_n$ ,

output:  $(s_1, s_2), r$ .

$\max r$

$$d = \left| \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}} \right|$$

$$\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}} \geq r \text{ - line above } P$$

$$- \frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}} \leq -r \text{ - line below } P$$

$\implies$  LP in 3 dim.

Note:  $\frac{s_2 - a_i s_1 - b_i}{\sqrt{a_i^2 + 1}}$  positive if  $(s_1, s_2)$  above the line, negative otherwise.



# Poglavje 10

## Hashing

A hash function is a random function,

$$h : U \rightarrow \{0, 1 \dots n - 1\} = M,$$

$U$  - universe,

$$u = |U|,$$

$$m = |M|.$$

Ideally we would like for  $h$  to be as completely random:  $P(h(x) = t) = \frac{1}{m}$ .

Standard application.

Let  $V \subset U$ ,  $|V| \ll |U|$ .

We would like to quickly answer if  $x \in V$  for every  $x \in V$ .

Solution:

- take  $h : U \rightarrow M$ ,
- make a table  $T = [0, 1 \dots n - 1]$ ,
- for  $v \in V$ :

$$T[h(v)] = 1,$$

$$T[y] = 0 \quad \forall y \in h(V).$$

- Let  $x \in V$ . Check

- if  $T[h(x)] = 1 : x \in V$ ,
- else:  $x \notin V$ .

Note: this is not OK:  $h$  not injective.

For  $x \in U$ , tell if  $x \in V$  in  $O(1)$ .

$h = \text{SHA256} : U \rightarrow \{0, 1\}^{256}$ .

Approach:

- design a family of hash functions,
- study collisions  $P_h(h(x) = h(y))$ ,
- $H$  needs to be „simple“.

Bad example:  $H =$  all functions from  $U$  to  $M$  storing  $h \in H$  would take  $|U| \log_2 |M|$  bits.

**Definicija 10.0.1.** A family of hash functions to be universal if for  $\forall x, y \in U, x \neq y, h \in H : P(h(x) = h(y)) \leq \frac{1}{m}$  (probability of collision).

$k$ -independent if  $\forall x_1 \dots x_k \in U$  pairwise different,  $\forall t_1 \dots t_k \in M P_r(h(x_i) = t_i \forall i) \leq \frac{1}{m^k}$ .

*Primer.*

$U = \{0, 1, 2, 3\}$ ,

$M = \{0, 1\}$ ,

$H = \{h_0, h_1, h_2\}$ ,

$h_0 : \{0 \rightarrow 0, 1 \rightarrow 0, 2 \rightarrow 1, 3 \rightarrow 1\}$ ,

$h_1 : \{0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 0, 3 \rightarrow 1\}$ ,

$h_2 : \{0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1, 3 \rightarrow 0\}$ .

$P(h(0) = h(2)) = \frac{2}{3} < \frac{1}{2}$  - not universal.

Why universal?

$U, V, M$

$H$  universal:  $\forall x, y : P(h(x) = h(y)) \leq \frac{1}{m}$ .

$X$ : number of collisions of  $V$ .

$$E(X) = E\left(\sum_{x,y \in V, x \neq y} X_{x,y}\right)$$

$$X_{x,y} = \begin{cases} 1 & \text{if } h(x) = h(y) \\ 0 & \text{else} \end{cases}$$

$$E(X) = \sum_{x,y \in V, x \neq y} E(X_{x,y}) \leq \binom{n}{2} \cdot \frac{1}{n}.$$

$U, V, M, H$

$T[0 \dots m-1]$

$\forall v \in V$

$T[h(v)] = v.$

For  $x \in V$  we check  $T[h(x)]$  if equals  $x$ ,

for  $y \in U \setminus V$ ,  $T[h(y)] \neq y$ .

For  $z \in V$ ,  $T[h(z)]$  can happen  $\neq z$  if  $h$  has collisions in  $V$ .

**Lema 10.0.2.** Let  $m \geq n^2$  and  $H$  universal. Then the probability that  $h$  has no collisions in  $V \geq \frac{1}{2}$ .

**Dokaz 10.0.3.**

$X$ : number of collisions

$$E(X) \leq \binom{n}{2} \cdot \frac{1}{m} < \frac{n^2}{2} \cdot \frac{1}{n^2} = \frac{1}{2}$$

$$P(X \geq 1) \stackrel{\text{Markov}}{\leq} \frac{E(X)}{1} = \frac{1}{2}$$

$$P(X = 0) \geq \frac{1}{2}.$$

*Primer* (Universal hash family).

$U = \{0, 1 \dots u-1\}$  (bits  $\equiv$  numbers)

$M = \{0, 1 \dots m-1\}$ .

Define: let  $p \geq u$ ,  $p$  prime number.

Define for  $a, b \in \mathbb{Z}_p$ ,  $a \neq 0$ .

$$h_{a,b} = (ax + b) \bmod m$$

$$ax + b \in \mathbb{Z}_p$$

$$H = \{h_{a,b} \mid a, b \in \mathbb{Z}_p, a \neq 0\}.$$

**Dokaz 10.0.4.**  $P(h_{a,b}(x) = h_{a,b}(y)) = ?$

$x, y$  fixed.

For any  $a, b$  denote

$$ax + b = t_x$$

$$ay + b = t_y :$$

$$a \sqcup + b \in \mathbb{Z}_p.$$

$$\begin{bmatrix} x & 1 \\ y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

$$\det \begin{bmatrix} x & 1 \\ y & 1 \end{bmatrix} \neq 0, \text{ because } x \neq y$$

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} x & 1 \\ y & 1 \end{bmatrix}^{-1} \begin{bmatrix} t_x \\ t_y \end{bmatrix}.$$

For each  $t_x, t_y$  there exists 1  $a, b$  mapping to  $t_x, t_y$ .

$$h_{a,b}(x) = h_{a,b}(y) \iff t_x = t_y \pmod{m}.$$

This holds for  $p \left( \lceil \frac{p}{m} \rceil + 1 \right)$

$p$ : choice of  $t_y$

$$t_x = t_y + km$$

$$P(h_{a,b}(x) = h_{a,b}(y)) \leq \frac{p(\lceil \frac{p}{m} \rceil - 1)}{p(p-1)} \leq \frac{\frac{p-1}{m}}{p-1} = \frac{1}{m}.$$

Function random for 2 elements, fixed for  $\geq 3$ .

Higher k-independent: better.

## 10.1 Chaining

$$V, U, h : U \rightarrow V.$$

Answer  $x \in V$  in  $O(1)$ .

$$T[0 \dots m-1]$$

$$n = |V|$$

$$\forall v \in V:$$

$$h(v_1) = h(v_2) \rightarrow [v_1 \ v_2 \dots] \text{ - linked list.}$$

Now:

$$x \in U.$$

Check if  $x$  is in list at  $T[h(x)]$ .

Check takes  $O(\text{length of a list at } h(x)) = 1 + \text{number of collisions with } x$ .

$X_x$ : number of collisions with  $x$ .

$E(X_x) = \sum_{y \in V} E(X_{x,y}) \leq n \cdot \frac{1}{m}$  if hash function is universal.

$\alpha = \frac{n}{m}$ : load factory (how many elements in 1 place).

$E(X_x) = 1$

$E(\max_x X_x) \neq \max_x E(X_x) = 1$ .

**Izrek 10.1.1.** Assume we throw  $n$  balls into  $n$  bins uniformly at random.

Then with high probability the fullest contains  $\theta\left(\frac{\log n}{\log(\log n)}\right)$  balls.

**Dokaz 10.1.2.**

$$\stackrel{?}{\leq} \frac{3 \ln n}{\ln \ln n}.$$

Let  $X_j$  be the number of balls in bin  $j$ .

$P\left(X_j \geq \frac{3 \ln n}{\ln \ln n}\right) = P(\text{there exists subset } S \text{ of balls thrown to bin } j).$

$|S| = k$

$$\begin{aligned} & P\left(\bigcup_{S \text{ balls}, |S|=k} \text{balls from } S \text{ are thrown to bin } j\right) \\ & \leq \sum_{S \text{ balls}, |S|=k} P(\text{balls from } S \text{ are thrown to } j) \\ & = \binom{n}{k} \left(\frac{1}{n}\right)^k \\ & \leq \frac{n^k}{k!} \cdot \frac{1}{n^k} = \frac{1}{k!} = (*). \end{aligned}$$

Note:  $e^x = \sum_{i=1}^{\infty} \frac{k^i}{i!} \geq \frac{k^k}{k!}$ .

$$\begin{aligned} (*) & \leq \frac{e^k}{k^k} \\ & = \left(\frac{e \ln n}{3 \ln \ln n}\right)^{\frac{3 \ln n}{\ln \ln n}} \\ & \leq e^{\frac{3 \ln n}{\ln \ln n} \cdot (\ln \ln \ln n - \ln \ln n)} \\ & = e^{-3 \ln n + \frac{\ln \ln \ln n \cdot (\ln n \cdot 3)}{\ln \ln n}} = (**). \end{aligned}$$

$$\frac{\ln \ln \ln n}{\ln \ln n} \rightarrow 0$$

$$(**) \leq e^{-3 \ln n + \ln n} = \frac{1}{n^2}.$$

$P(\text{at least for 1 bin } j \geq k) = n \cdot \frac{1}{n^2} = \frac{1}{n}.$

$U, V, H$  hash family,  $h : U \rightarrow M$

$v \in V$

$n = |V|$

max load  $O\left(\frac{\log n}{\log(\log n)}\right)$ .

Perfect hashing: we would like

- $O(1)$  lookup (worst case)
- $O(n)$  size of table.

## 10.2 2 level hashing

Input:  $V$

$n = |V|$ .

Take hash function from universal family with  $m = |M| = n$ .

Count total collisions  $X$ .

$$E(X) \leq \binom{n}{2} \cdot \frac{1}{m} \leq \frac{n}{2}$$

$$P(x \geq n) \stackrel{\text{Markov}}{\leq} \frac{1}{2}$$

$\implies$  by repeating sample  $h$  we can guarantee

- for each  $i \in M$  we store at  $T[i]$  another hash table of size  $C_i^2$ , where  $C_i$  = number of elements of  $V$ , hashed in  $i$ ,
- we sample  $h_i$  from universal hash family with  $M_i = C_i^2$ .

$P(h_i \text{ has no collisions}) \geq \frac{1}{2}$  (by lemma).

We resample if  $h_i$  has collisions.

$$E(\text{sampling } h_i) = 2.$$

Construction time:

- step 1:  $O(n)$
- step 2:  $O(C_1 + \dots + C_n) = O(n)$ ;

together  $O(n)$ .

Lookup time:  $O(1)$  (evaluating  $h(x)$  and  $h_{h(x)}(x)$ ).

Space:  $O(C_1^2 + \dots + C_n^2)$  in  $O(n)$ .

By first step  $n > \text{number of collisions of } h = \sum_{i=1}^n \binom{C_i}{2} = \sum_{i=1}^n \frac{C_i^2 - C_i}{2}$

$\implies \sum_{i=1}^n C_i^2 < 2n + \sum_{i=1}^n C_i = 3n$ .

### 10.3 The power of 2 choices

Variant: placing  $n$  balls in  $n$  bins but for each ball we choose  $d$  balls uniformly at random and put the ball in bin with minimal load.

**Izrek 10.3.1.** The above process with  $d \geq 2$  results in at most maximum load of  $O\left(\frac{\ln(\ln n)}{\ln d}\right)$ .

**Dokaz 10.3.2.** (sketch).

$b_i$  = upper bound of the number of bins with load at most  $i$ .

Height of a ball = the number of balls in the bin, where the ball is placed.

$P(\text{a ball has height at least } i+1) \leq \left(\frac{b_i}{n}\right)^d$  (choose  $d$  times independently).

$X^{i+1}$ : number of balls with height  $\geq i+1$ .

$$X^{i+1} = \sum_{j=1}^n X_j^{i+1}$$

$X_j^{i+1}$ : indicator variable of  $j$ -th ball having height  $i+1$ .

$$E(X^{i+1}) \leq \sum_{j=1}^n \left(\frac{b_i}{n}\right)^d = d \cdot \left(\frac{b_i}{n}\right)^d.$$

Chernoff bound: with high probability  $X^{i+1} \leq 2n \left(\frac{b_i}{n}\right)^d$ .

$X^{i+1} \geq$  number of bins with load at least  $i+1$ .

Define (set)

$$b_{i+1} = \sum \frac{b_i^d}{n^{d-1}}$$

$$b_4 = \frac{n}{4}$$

$$b_{i+4} = \frac{n}{2^{2 \cdot d^i - \sum_{j=0}^{i-1} d^j}}$$

$$i = 0: b_4 = \frac{n}{2^{2^1}} = \frac{n}{4}$$

$i \rightarrow i + 1$ :

$$\begin{aligned}
 b_{i+4} &= \frac{2 \cdot b_{i+3}}{n^{d-1}} \\
 &\stackrel{IH}{=} \frac{2 \cdot \left( \frac{n}{2^{2 \cdot d^i - \sum_{j=0}^{i-1} d^j}} \right)^d}{n^{d-1}} \\
 &= \frac{2^1 \cdot n^d}{n^{d-1} \cdot 2^{2 \cdot d^{i+1} - \sum_{j=1}^i d^j}} \\
 &= \frac{n}{2^{2 \cdot d^{i+1} - \sum_{j=0}^i d^j}}.
 \end{aligned}$$

In particular:  $b_{i+4} \leq \frac{n}{2^{d^i}} < 1$  when?

$$n < 2^{d^i}$$

$$\log_2 n < d^i$$

$$\log_d \log_2 n < i$$

$$\implies \text{for } i = \frac{\log(\log_2 n)}{\log d} \text{ is } b_i < 1 \implies \text{no bins with load } > \frac{\log(\log_2 n)}{\log d}.$$

Application:

We sample 2 hash functions  $h_1, h_2 : U \rightarrow M$ .

For element  $v \in V$  we insert in  $T[h_1(v)]$  or  $T[h_2(v)]$  depending on which list is shorter.

Max load in  $O(\log(\log n))$ .

## 10.4 Cockoo hashing

Idea: use 2 hash functions but allow moving elements later.

We want to have at most 1 element at each entry in the table.

Inserting:

- if empty: insert,
- if not empty: push other element to its other choice, repeat recursively.



Questions:

- how many do I need to move,
- how many elements can I insert before problems?

We can think of positions in the table as vertices and elements of  $V$  as edges.  $|V|$  edges are inserted uniformly at random (if ideal hash function)  $\implies$  random graph.

Erdős-remy model:  $G_{n,m} \approx G_{n,p}$  if  $m = \binom{n}{2}p$  (A.S. properties).

If  $np < 1 - \epsilon$ : all connected components have size at most  $O(\log n)$ , components are trees or at most 1 cycle per component, expected size of a component is  $O(1)$ .

Fact: if graph has at most 1 cycle per component, then inserting can be done and takes at most  $2 \cdot (\text{size of component})$  time (each edge changes direction at most 2 times).

#### **Izrek 10.4.1.**

Let  $n = |U|$ ,  $h_1, h_2 : U \rightarrow M$ ,  $m = |M| = 2 \cdot (1 + \epsilon) \cdot n$ , then with high probability cuckoo hashing works correctly with

- inserting time:
  - $O(\log n)$  time worst case,
  - $O(1)$  expected case,
- space:  $O(n)$ ,
- lookup time:  $O(1)$ .

Dynamically add element:

$$m = 2 \cdot (1 + \epsilon) \cdot n$$

$$p = \frac{m'}{\binom{n'}{2}} = \frac{2m'}{n'(n'-1)}$$

$$pn' = \frac{2m'}{(n'-1)} = \frac{2n'}{2(1+\epsilon)n'} = \frac{1}{1+\epsilon} < 1 + \epsilon'$$

## 10.5 Bloom filter

Take  $k$  hash functions  $h_1 \dots h_k$  at random,  $h_i : U \rightarrow M, T[0 \dots m - 1]$ .

$V \subset U$ , for every element  $v \in V$  set  $T[h_i(v)] = 1 \forall i \in \{1 \dots k\}$ .

False positives:  $x \notin V$  such that  $T[h_i(x)] = 1 \forall i \in \{1 \dots k\}$ .

For each  $T[j]$   $P(T[j] = 0) = \left(1 - \frac{1}{m}\right)^n \approx e^{-\frac{nk}{m}}$ ;

$k$ : each hash function,  $n$ : for each  $v$ .

Now

$P(T[h_i(x)] = 1 \forall i, \forall x \notin V) \approx \left(1 - e^{-\frac{nk}{m}}\right)^k = f(k)$  - probability of a false positive.

$\left(1 - e^{-\frac{nk}{m}}\right)$ : 1 position.

Searching for a minimum:

$$f'(k) = 0$$

$$\implies k = \ln 2 \cdot \frac{m}{n}$$

$$f\left(\ln 2 \cdot \frac{m}{n}\right) = \left(\frac{1}{2}\right)^{\ln 2 \cdot \frac{m}{n}} \approx 0.6185^{\frac{m}{n}}$$

$\implies$  we choose  $m$  such that  $0.6185^{\frac{m}{n}}$  small (in  $O(n)$ )

$\implies$  calculating  $k = \ln 2 \cdot \frac{m}{n}$

$\implies$  hashing with space  $O(n)$

$\implies$  checking in  $O(1)$

$\implies$  probability of error small.

## 10.6 Linear probing

$V \subset U, h : U \rightarrow M, T[0 \dots m - 1]$ .

- Insert  $v \in V$ : check  $T[h(v)], T[h(v) + 1], T[h(v) + 1] \dots$  until finding empty space, then insert it.
- Check if  $x \in V$  by checking  $T[h(x)] \stackrel{?}{=} x, T[h(x) + 1] \stackrel{?}{=} x \dots$  until finding  $x$  or finding empty.

$x \in U$

$X$ : number of steps to check if  $x \in V$ .

$E(X) = ?$

Block of size  $2^l$  is bad if it has more than  $2^l \cdot \frac{2}{3}$  values.

Set  $\frac{n}{m} = \frac{1}{3}$ .

Expected number of elements hashed in block of size  $2^l$  is  $\frac{1}{3} \cdot 2^l$ .

$$\begin{aligned} E(X) &= \sum_{i=0}^n P(X = i) \cdot i \\ &\leq \sum_{j=0}^{\log_2 n} P(2^{j-1} < X \leq 2^j) \cdot 2^j \\ &\leq \sum_{j=0}^{\log_2 n} P(\text{block above } h(x) \text{ of size } 2^j \text{ is bad}) \cdot c \cdot 2^j. \end{aligned}$$

$c$ : not aligned?

$P(\text{block of size } 2^j \text{ is bad}) = P(Y > \frac{2}{3} \cdot 2^j) = P(Y - \frac{1}{3} \cdot 2^j > \frac{1}{3} \cdot 2^j);$

$Y$ : number of elements hashed to the block.

$E(Y) = \frac{1}{3} \cdot 2^j$

$E(X) \stackrel{\text{Chernoff}}{\leq} e^{-k \cdot 2^j}$ ; Chernoff: sum of independent indicators.

$E(X) < O(1) \cdot \sum_{j=0}^{\log_2 n} 2^j \cdot e^{-k \cdot 2^j}$  in  $O(1)$

$\implies$  checking in  $O(1)$ .

Chernoff: if ideal hash function; 5 independent is enough.

# Poglavje 11

## Data streams

Stream of values

$$\sigma = a_1, a_2 \dots a_n$$

$a_i$ : tokens

$$a_i \in [n]$$

$m$ : length of stream (very large).

**Definicija 11.0.1.**  $f_i = |\{j \mid a_j = i\}|$

We could be interested in

- number of different token,
- frequency of some token,
- frequent tokens:  $\{i \in [n] \mid f_i \geq \frac{m}{10}\}$
- moments:  $\|f\|^2 = \sum_{i \in [n]} f_i^2$
- $\vdots$

We want to use memory in  $O(\text{poly}(\log n, \log m)) \ll O(n, n)$ .

Most problems cannot be solved precisely, hence we search for  $(\epsilon, \delta)$ -approximation.

Algorithm  $A(G)$ :

- initialization,

- incremental steps,
- finalization

using randomness (oblivious stream - it doesn't know which randomly, e.g. we can choose stream that „attacks algorithm“).

## 11.1 Count min sketch

For a given  $i \in [n]$  (token) at the end of stream give  $f_i$ .

$A(\sigma, \epsilon, \delta)$ :

Init:  $k = \lceil \frac{2}{\epsilon} \rceil, t = \lceil \log_2 \left( \frac{1}{\delta} \right) \rceil$ .

We choose  $t$  hash functions  $h_1 \dots h_t : [n] \rightarrow M = [k] = \{1 \dots k\}$  from a universal family  $H$ .

Let  $C[0 \dots t-1][0 \dots k-1]$  be 2-dim (hash) table,  $C[i][j] = 0 \forall i, j$ .

Updates:

for every token  $a_i \in \sigma$  we update  $C$

for  $j = 0, 1 \dots t-1$

$$C[j][h_j(a_i)] += 1$$

Output: we asked  $a \in [n]$ , return  $\overline{f}_a = \min_{0 \leq j \leq t-1} C[j][h_j(a)]$ ; min collisions.

### Izrek 11.1.1.

For every  $a \in [n]$  it holds

$$f_a \leq \overline{f}_a \leq f_a + \epsilon m$$

with probability at least  $1 - \delta$ .

Notice: space needed  $O(t \cdot k \cdot \log m) = O\left(\frac{2}{\epsilon} \cdot \log_2 \left(\frac{1}{\delta}\right) \log m\right)$ .

### Dokaz 11.1.2.

$$\forall i \in [t] : C[i][h_i(a)] \geq f_a \implies \overline{f}_a \geq f_a.$$

Fix  $a$ .

Let  $X_i = C[i][h_i(a)] - f_a$  excess of  $i$ -th count.

$$I_{x,y}^i = \begin{cases} 1 & \text{if } h_i(x) = h_i(y) \\ 0 & \text{else} \end{cases}$$

$$X_i = \sum_{y \in [n], y \neq a} I_{x,y}^i \cdot f_y.$$

$$\begin{aligned} E(X_i) &= \sum_{y \in [n], y \neq a} E(I_{x,y}^i) \cdot f_y \\ &\stackrel{*}{\leq} \sum_{y \in [n], y \neq a} \frac{1}{k} \cdot f_y \\ &\leq \frac{1}{n} \cdot m \\ &\stackrel{**}{\leq} \frac{m}{2} \end{aligned}$$

\*: hash function from universal family.

\*\* :  $P(X_i \geq \epsilon m) \stackrel{\text{Markov}}{\leq} \frac{\epsilon m}{2\epsilon m} = \frac{1}{2}$  for fixed  $i$ .

$$\begin{aligned} P(\overline{f_a} - f_a \geq \epsilon m) &\leq P(X_i \geq \epsilon m \ \forall i) \\ &\stackrel{\text{indep.}}{=} \left(\frac{1}{2}\right)^t \leq \delta. \end{aligned}$$