Advanced Algorithmics

Strategies for Tackling Hard Problems

Sebastian Wild

Markus Nebel

Lecture 15

2017-06-08

Universal Hashing – Efficient Randomized Hashing

Balls-into-bins model is optimistic!

It assumes that $B_1, \dots, B_n$ are mutually indep.

B_j = the bin of jth ball

$B_j \in [m]$

$|S| = n$ $|U|$

For fully random hash function $h: S \to [m]$ we need $\geq ld(m^n)$

$= n \, ld(m)$

bits to distinguish all functions

m^n different hash functions

$\longrightarrow$ too expensive

Definition 4.51 (Universal Family)
Let $\mathcal{H}$ be a set of hash functions from U to R with $|R| = m$ and $|U| \geq m$. Assume $h \in \mathcal{H}$ is chosen uniformly at random.
Then $\mathcal{H}$ is called a *universal* if

implies

$$\forall x_1, x_2 \in U \ : \ x_1 \neq x_2 \ \rightarrow \ \Pr[h(x_1) = h(x_2)] \ \leq \ \frac{1}{m}.$$

→ does not state anything about $\Pr[h(x_1) = h(x_2) \wedge h(x_1) = h(x_3)]$

$\mathcal{H}$ is called *strongly universal* or *pairwise independent* if

$h(x_1), h(x_2), \dots h(x_n)$ pairwise indp.

$$\forall x_1, x_2 \in U, y_1, y_2 \in R \ : \ x_1 \neq x_2 \ \rightarrow \ \Pr[h(x_1) = y_1 \wedge h(x_2) = y_2] \ \leq \ \frac{1}{m^2}.$$

Examples:

$h_{ab}(x) = ((a \cdot x + b) \bmod p) \bmod m$

p prime $\quad p \geq m$

$U = [0..u]$

- $\mathcal{H}_1 = \{ h_{ab} : a \in [1..p), b \in [0,p] \}$ universal
- $\mathcal{H}_2 = \{ h_{ab} : a, b \in [0..p-1] \}$ strongly universal
- $h_a(x) = (ax \bmod 2^k) \operatorname{div} 2^{k-\ell}$ $\quad U = [0..2^k)$

$\mathcal{H}_3 = \{ h_a : a \in [1..2^k), a \text{ odd} \}$ universal $\quad R = [0..2^\ell)$

How good is universal hashing?

Define $X_{ij} = [x_i \text{ and } x_j \text{ land in same bin}]$

$$X = \sum_{1 \le i < j \le n} X_{ij} = \#\,\text{collisions}$$

$h \in \mathcal{H}$ from universal class chosen uniformly at random

⚠ $\Pr[X_{ij} = 1] \le \frac{1}{m}$

but X_{ij}, X_{ik} not necessarily independent

$$\mathbb{E}[X] = \sum_{1 \le i < j \le n} \mathbb{E}[X_{ij}] \le \frac{1}{m} \cdot \underbrace{\binom{n}{2}}_{\frac{n(n-1)}{2}} < \frac{n^2}{2m}$$

$Y = \#$ balls in fullest bin

(bins & balls: $n = m$: $Y = \Theta(\log n / \log\log n)$ whp)

$$\rightsquigarrow X \ge \binom{Y}{2} = \frac{Y(Y-1)}{2}$$

$$\rightsquigarrow \Pr\left[Y \ge n\cdot\sqrt{\frac{2}{m}}\right] \le \Pr\left[X \ge \frac{n^2}{m}\right] = \Pr[X \ge 2\cdot\mathbb{E}[X]] \underset{\text{Markov}}{\le} \frac{1}{2}$$

$n = m \quad \rightsquigarrow \quad Y \leq \sqrt{2n}$ with prob $\geq \frac{1}{2}$

Nevertheless often sufficient performance in practice

- defense against worst cases
- typical inputs have themselves some randomness

Perfect Hashing: Random Sampling

Static Hashing: $S \subseteq U$ fixed $|S| = n$

→ build data structure with $O(1)$ search / access

no insert / delete

→ $O(n)$ space

If $h \in \mathcal{H}$ chosen uniformly at random

universal

If $m \geq n^2$ $\Pr[X \geq 1] \leq \Pr[X \geq \frac{n^2}{m}] = \Pr[X \geq 2\,\mathbb{E}[X]] \leq \frac{1}{2}$

$\Downarrow$

$\Pr[\text{no collisions}] = \Pr[X = 0] \geq \frac{1}{2}$

fine, but $\Theta(n^2)$ space is too much.

Can we improve? Yes:

$Y_1, \dots, Y_n$ number balls in h_1-bins

$\rightsquigarrow$ space: $n + \sum_{j=1}^{n} y_j^2$

for bins

(2) h_1 chosen by rejection sampling

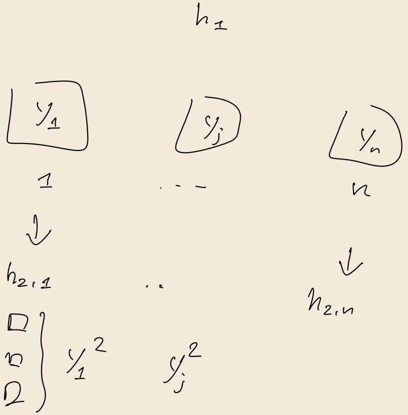
until $X \leq n$ (h_1 $n = m$)

$$\Pr\left[X \geq \frac{n^2}{m}\right] = \Pr[X \geq n] \leq \frac{1}{2}$$

y_j balls in bin j

$\rightsquigarrow \binom{y_j}{2}$ collisions

$$\sum_{j=1}^{n} \binom{y_j}{2} \leq n$$



(1) $h_{2,j}$ can be chosen without producing collisions by rej. sampling

$\hookrightarrow$ no collisions on second level

$$\binom{\ell}{2} = \frac{\ell(\ell-1)}{2} = \frac{\ell^2}{2} - \frac{\ell}{2} \quad \Rightarrow \quad \ell^2 = 2\binom{\ell}{2} + \ell$$

$$\Rightarrow \text{ space : } \quad n + \sum_{j=1}^{n} y_j^2 = n + 2\underbrace{\sum_{j=1}^{n}\binom{y_j}{2}}_{\leq n} + \underbrace{\sum_{j=1}^{n} y_j}_{\leq n} \leq 4n$$

$$\Rightarrow \Theta(n) \text{ space}$$

<u>Remarks</u> Can make this dynamic $\leadsto$ $O(1)$ amortized expected time insert & delete

Random Sampling: Local Search

Warmup: A randomized 2SAT algorithm

procedure localSearch2SAT(ϕ, *certainty*):
  k = number of variables of ϕ
  Choose assignment $\alpha \in \{0,1\}^k$ uniformly at random. ← not necessary
  for $j = 1, \ldots, certainty \cdot 2k^2$
    if α fulfills ϕ **return** "ϕ satisfiable"
    Arbitrarily choose a clause $C = \ell_1 \vee \ell_2$ that is not satisfied under α. ← not random
    Choose ℓ from $\{\ell_1, \ell_2\}$ uniformly at random. random!
    α = assignment obtained by negating ℓ.
  return "ϕ probably not satisfiable"

Theorem 4.52 (localSearch2SAT is OSE-MC for 2SAT)

Let ϕ be a 2SAT formula.

1. If ϕ is unsatisfiable, `localSearch2SAT` always returns "probably not satisfiable".
2. If ϕ is satisfiable, `localSearch2SAT` returns "satisfiable" with probability at least $1 - 2^{-certainty}$.

Proof: (1) trivial from code

(2) ϕ satisfiable $\rightsquigarrow$ α^* that satisfies ϕ

Define $\alpha_1, \alpha_2, \ldots$ as assignment in step $j = 1, 2, \ldots$

$X_i = k - d_H(\alpha_i, \alpha^*)$ = # matching variable assignments

$X_i = k$ $\rightsquigarrow$ terminates

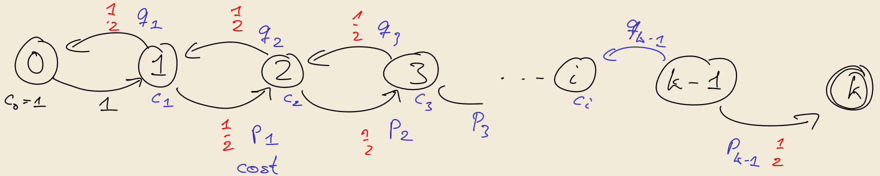
otherwise $\rightsquigarrow$ $X_{i+1} \in \{X_i - 1, X_i + 1\}$

$\Pr[X_{i+1} = 1 \mid X_i = 0] = 1$

$\Pr[X_{i+1} = j+1 \mid X_i = j] \overset{=}{\geq} \frac{1}{2}$ $\quad$ l_1 or l_2 must differ between α_i and α^*

$\Pr[X_{i+1} = j-1 \mid X_i = j] \overset{=}{\leq} \frac{1}{2}$

upper bound #step till $X_i = k$ by Markov chain



y_i = expected # steps till we reach (k)

$p_i = 1 - q_i$

$y_k = 0$

$y_0 = 1 + y_1$

$= 1$

$y_i = C_i + p_i \cdot y_{i+1} + q_i y_{i-1}$ $\qquad 1 \le i \le k-1$

$\parallel$

$p_i y_i + q_i y_i \qquad \Longleftrightarrow \quad p_i (y_{i+1} - y_i) = q_i (y_i - y_{i-1}) - c_i$

$\Longleftrightarrow \quad \underbrace{y_{i+1} - y_i}_{\dot{y}_i} = a_i \underbrace{(y_i - y_{i-1})}_{\dot{y}_{i-1}} + b_i \qquad a_i = \frac{q_i}{p_i}$

$b_i = -\frac{c_i}{p_i}$

$\dot{y}_i = a_i \dot{y}_{i-1} + b_i$

$\dot{y}_0 = y_1 - y_0 = -1 = -c_0$

$$\dot{y}_i = \left(\prod_{j=1}^{i} a_j\right) \cdot \underset{\underset{-1}{\shortparallel}}{\dot{y}_0} + \sum_{j=1}^{i} \left(\prod_{k=j+1}^{i} a_k\right) b_j$$

$$\sum_{\ell=0}^{i-1} \underset{\underset{y_{\ell+1}-y_\ell}{\shortparallel}}{\dot{y}_\ell} = y_i - y_0 \implies \underset{\underset{0}{\shortparallel}}{y_k} = y_0 + \sum_{\ell=0}^{k-1} \dot{y}_\ell$$

$$\implies y_0 = -\sum_{\ell=0}^{k-1} \dot{y}_\ell$$

$$a_i = a = \frac{q_i}{p_i} = 1 \qquad b_i = b = -\frac{c_i}{p_i} = -2$$

$$\dot{y}_i = -1 - 2\sum_{j=1}^{i} 1 = -2i - 1$$

$$y_0 = \sum_{\ell=0}^{k-1} (2\ell + 1) = k^2 \implies \mathbb{E} \text{ iterations of loop is } k^2$$

$$\Pr[\#\text{iterations} \geq \text{certainty} \cdot 2k^2] \underset{\text{Markov}}{\leq} \frac{1}{\text{certainty} \cdot 2}$$

$$\Pr[\#\text{iterations} \geq 2k^2] \leq \frac{1}{2}$$

certainty independent repetitions $\rightsquigarrow$ $\left(\frac{1}{2}\right)^{\text{certainty}}$ $\square$.