Reading: Text:

Hashing, Randomized Rounding

Linear Probing

[[*See STOC'07 paper of Pagh, Pagh, Ruzic.*]]

Note: Analysis for $b = 3n$ to ease notation.

Consider binary tree spanning array of buckets:

- leaves level 0
- node at level k has 2^k array positions under it
- expect node of level k to have $(1/3)2^k$ items hashed to buckets under it

  [[*In sense of original location $h(x)$, not $h(x)+1$, $h(x)+2$, etc.*]]

Def: A node of level k is *dangerous* if more than $(2/3)2^k$ elts hash under it.

To bound operation time, must bound size of contiguous run of elts. containing $h(q)$:

Claim: If $2^k \leq$ size of run $\leq 2^{k+1}$, either $(k-2)$-ancestor of $h(q)$ or a nearby sibling is dangerous.

Proof: Counting argument.

Let E_k be event a level-k node is dangerous. Expected operation time:

$$\sum_k O(2^k)\Pr[2^k \text{run}(h(q)) \leq 2^{k+1}] \leq \sum_k O(2^k)\Pr[E_{k-2}].$$

Balls-in-bins: want to bound prob. bin of expected size $\mu = n/b = 2^k/3$ has more than 2μ balls

- Markov: $\Pr[X \geq 2\mu] \leq 1/2$, exp. diverges
- Chebyshev:

$$\begin{aligned}\Pr[X \geq 2\mu] &= \Pr[(X-\mu) \geq \mu] \\ &\leq \Pr[(X-\mu)^2 \geq \mu^2] \\ &\leq E[(X-\mu)^2)]/\mu^2 \\ &= O(1/\mu)\end{aligned}$$

- 4th moment:

  First compute moment. Let $Y_i = X_i - (1/b)$ where X_i indicates ith ball in bin, so $E[Y_i] = 0$.

$$E[(X-\mu)^4] = E[(\sum Y_i)^4] = \sum E[Y_iY_jY_kY_l]$$

  - one index, say i, appears once: $E[Y_iY_jY_kY_l] = E[Y_i]E[Y_jY_kY_l] = 0$
  - all equal: $E[Y_i^4] = O(1/b)$
  - two pairs: $E[Y_i^2Y_j^2] = E[Y_i^2]E[Y_j^2] = O(1/b^2)$

  so $E[(X-\mu)^4] = O(n/b + (n/b)^2) = O(n^2/b^2) = O(\mu^2)$, and

$$\Pr[X \geq 2\mu] = \Pr[(X-\mu) \geq \mu]$$

$$\begin{aligned} &\leq \Pr[(X-\mu)^4 \geq \mu^4] \\ &\leq E[(X-\mu)^4)]/\mu^4 \\ &= O(1/\mu^2) \end{aligned}$$

[[*Why not 3rd moments? Get negatives so can't apply Markov.*]]

By 4th moment, expected operation time at most $\sum_k O(2^k)O(2^{-2k}) = O(1)$.

Cuckoo Hashing

Idea: Place n keys into two arrays and resolve collisions by bumping to other array.

- two arrays $A[1..b]$ and $B[1..b]$, where $b = 2n$
- two hash functions h and g
- when x arrives, if $A[h(x)]$ contains elt y, recursively tro to move y to $B[g(y)]$

Note: Think random bipartite graph, nodes array positions, edges $(h(x), g(x))$, edge probability n/b^2

Analysis:

- hashing succeeds (no cycles): show constant prob. of collision
- fail: then rehash, must bound prob. of cycles

No cycles

$$\begin{aligned} \Pr[1st \text{ evict}] &= \sum_y \Pr[h(x) = h(y)] \\ &= n/b \\ &= 1/2 \end{aligned}$$

Pr[lth evict] at most 2^-l by induction, so expected running time is $\sum_l l \cdot 2^{-l}$, constant.

Rehashing

- Prob. fixed cycle of length l:

$$(n/b^2)^l$$

- # cycles of length l:

$$b^l$$

- Prob. exists cycle of length l:

$$(n/b)^l = 2^{-l}$$

- Prob. exists cycle:

$$\sum_l 2^{-l} = O(1)$$

[[*How much randomness do we need for these? STOC'07 says can cuckoo hash with pairwise independence!*]]

Randomized Rounding

Max-SAT

Def: A *satisfiability* formula consists of

- n Boolean variables x_i
- m disjunctive clauses C_i

Example: $(x_1 \wedge \neg x_2 \wedge x_3) \vee (x_3) \vee (\neg x_1 \wedge x_2)$

Note: Terminology: literal, length of clause, ...

Problem: MAX-SAT: Given weights w_i for clauses C_i, find assignment that maximizes value of satisfied clauses.

Question: Approximation?

- uniform random sampling:

  Claim: Let $x_i = 1$ w/prob. $p = 1/2$. This is a (1/2)-approximation.

  Proof: Let Y_j indicate if C_j is satisfied. Then

  $$E[\sum_j w_j Y_j] = \sum_j w_j \Pr[C_j = 1],$$

  and since $C_j = 1$ iff each literal is true,

  $$\Pr[C_j = 1] = (1 - (1/2)^{l_j}) \geq 1/2.$$

  Note: Better for longer clauses: optimal if $l_j = 3 \ \forall \ j$.

- biased random sampling:

  Claim: Let $x_i = 1$ w/prob. $p > 1/2$. Then $\Pr[C_j = 1] \geq \min(p, 1-p^2)$ if no negated unit clauses.

  Proof: Unit clauses ok since $p \geq (1-p)$. For clauses with a unnegated and b negated literals,

  $$\begin{aligned}\Pr[C_j = 1] &= 1 - p^a(1-p)^b \geq 1 - p^{a+b} \\ &= 1 - p^{l_j} \geq 1 - p^2.\end{aligned}$$

  Note: $p = 1 - p^2 \rightarrow p = \frac{1}{2}(\sqrt{5} - 1) \approx 0.618$

  Claim: Let $x_i = 1$ w/prob. $p > 1/2$. This is a p-approximation.

  Proof: Must show negated unit clauses don't hurt. Improve bound on opt:

  - assume WLOG weight v_i of $\neg x_i$ smaller than weight w_i of x_i
  - $OPT \leq \sum_j w_j - \sum_i v_i$

  Let U be clauses excluding negated ones. Note $\sum_{j \in U} w_j = \sum_j w_j - \sum_i v_i$. Count performance of alg only on clauses in U.

- randomized rounding:

  Idea: decouple the bias, use different bias for each variable.

LP Formulation

Variables:

- y_j for each variable
- z_j for each clause

Objective: $\max \sum_j w_j z_j$

Constraint: $\forall \ C_j, \ z_j \leq \sum_{i \in P_j} y_j + \sum_{i \in N_j} (1 - y_j)$

Rounding

Fact: Arithmetic-Geometric Mean Inequality: For non-negative a_i, $\prod_{i=1}^k a_i \leq ((1/k) \sum_{i=1}^k a_i)^k$.

Claim: Randomized rounding gives $(1 - 1/e)$-approx.

Proof: Want to bound prob. clause C_j of length l_j is satisfied. Let P_j be set of positive literals and N_j be set of negative literals and y^*, z^* be an optimal soln to the LP. Then

$$\begin{aligned}\Pr[C_j = 0] &= \prod_{i \in P_j} (1 - y_i^*) \prod_{i \in N_j} y_i^* \\ &\leq \left(\frac{1}{l_j} \sum_{i \in P_j} (1 - y_i^*) + \sum_{i \in N_j} y_i^* \right)^{l_j} \\ &= \left[1 - \frac{1}{l_j} \left(\sum_{i \in P_j} y_i^* + \sum_{i \in N_j} (1 - y_i^*) \right) \right]^{l_j} \\ &\leq \left(1 - \frac{z_j^*}{l_j} \right)^{l_j}\end{aligned}$$

where the first inequality is by arithmetic-geometric mean inequality and the second is from the constraint in the LP. Thus

$$\Pr[C_j = 0] \geq 1 - (1 - \frac{z_j^*}{l_j})^{l_j} \geq \left[1 - (1 - \frac{1}{l_j})^{l_j} \right] z_j^*$$

by concavity of function on unit interval and algebraic manipulation. The min. is for large l_j and approaches $(1 - 1/e)$ from above.