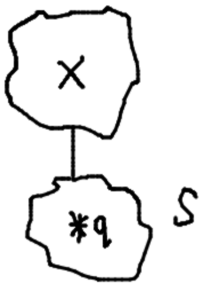
CSE 4502/5717 Big Data Analytics

Notes taken by: Yanting Chen

Lecture 6 – 02/12/2018

- 

  $X = k_1, k_2 \ldots k_n$

  Rank(q, s) = j

  Let Rank(q, X) = r_j

$$Prob\left(\left|r_j - j\frac{n}{s}\right| > \sqrt{3\alpha}\frac{n}{\sqrt{s}}\sqrt{logn}\right) \leq n^{-\alpha}$$

- To begin with, all the keys are alive, N=n;

  Repeat

  1. In one pass through the data pick a random sample S

     Each alive key will be in S with a prob. of $\frac{M}{2N}$

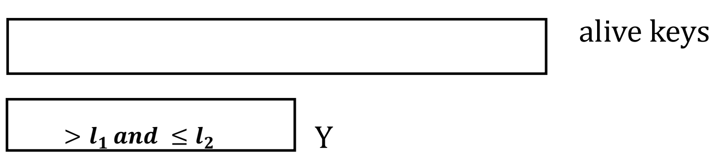
     $E[|S|] = \frac{M}{2}$

  2. Pick l_1 & l_2 from S such that,

     Rank$(l_1, S) = \ i \cdot \frac{s}{N} - \sqrt{4\alpha s \cdot \log N}$

     and Rank$(l_2, S) = \ i \cdot \frac{s}{N} + \sqrt{4\alpha s \cdot \log N}$

  3. Do one more pass through the data and store Y = {q: q is alive and $l_1 < q \leq l_2$} in the disk;



let $n_1 = |\{q: q \text{ is alive and } q < l_1\}|$;

$n_2 = |\{q: q \text{ is alive and } q \le l_2\}|$;

4. If $i < n_1$ or $i > n_2$

   or if $|Y| > \frac{N}{M^{0.4}}$ then start all over;

   else $i = i - n_1$ and $N = |Y|$;

   only the elements in Y are alive;

Until $N \le M$;

Identify and output the i^{th} smallest from the alive keys.

- ***Analysis:***

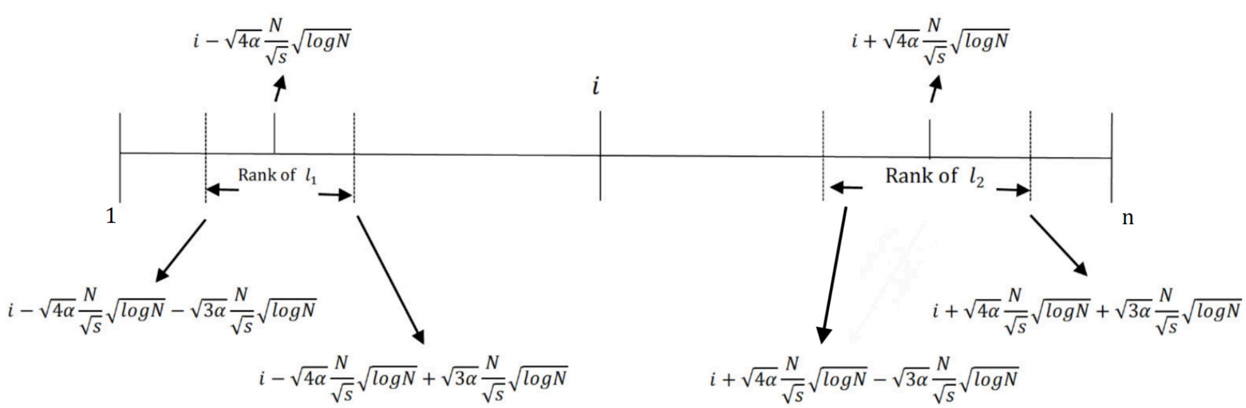
In step 1, the # of sample keys in S is Binomial with parameters N and $\frac{M}{2N}$

Using Chernoff bounds, $|s| = \tilde{\theta}(M)$

Expected Rank of l_1 in $X = [i \cdot \frac{s}{N} - \sqrt{4\alpha \cdot s \cdot \log N}]\,\frac{N}{s}$

$$= i - \sqrt{4\alpha}\frac{N}{\sqrt{s}}\sqrt{\log N}$$

Expected Rank of l_2 in $X = i + \sqrt{4\alpha}\,\frac{N}{\sqrt{s}}\sqrt{\log N}$



$$|Y| \leq i + (\sqrt{4\alpha} + \sqrt{3\alpha})\frac{N}{\sqrt{s}}\sqrt{\log N} - [i - (\sqrt{4\alpha} + \sqrt{3\alpha})\frac{N}{\sqrt{s}}\sqrt{\log N}]$$

$$\leq 2(\sqrt{4\alpha} + \sqrt{3\alpha})\frac{N}{\sqrt{s}}\sqrt{\log N} \text{ with a prob. of } \geq (1 - N^{-\alpha})$$

- ***w.h.p. (with high probability)***

$|Y| = O\left[(\sqrt{4\alpha} + \sqrt{3\alpha})\frac{N}{\sqrt{M}}\sqrt{\log N}\right]$ note: $N = M^c \to \log N = c \cdot \log M$

If N is a polynomial in M,

then $|Y| = O(\frac{N}{M^{0.4}})$ w.h.p.

- ***Example***

$M = 10^9$, $N \leq M^4$

I/O complexity:

$$\frac{2n}{B} + \frac{n}{M^{0.4}} \cdot \frac{2}{B} + \frac{2n}{M^{0.8}B} + \cdots \leq (2+\varepsilon)\frac{n}{B}, \ for\ any\ constant\ \varepsilon > 0$$

- ***A Graph Problem:***

Minimum spanning tree(MST)

Problem:

Input: a weighted, connected and undirected graph, G(V,E)

Output: A minimum spanning tree for G

- ***Prim's algorithm***

Grow a subtree by adding one edge at a time, starting with the lightest edge.

Let x be any node outside the tree

NEAR[x] = the closest tree neighbor of x;

1. pick the lightest edge e=(a,b) in E;
2. for every u ∈ V−{a,b} do
   if weight(u,a) < weight(u,b) then
   NEAR[u] = a; else NEAR[u]=b;
3. for u ∈ V−{a,b} do

insert u into a 2-3 tree Q where the key value is weight(u, NEAR[u]);

4. for i=1 to (n-2) do

find the node u in Q with the least key;

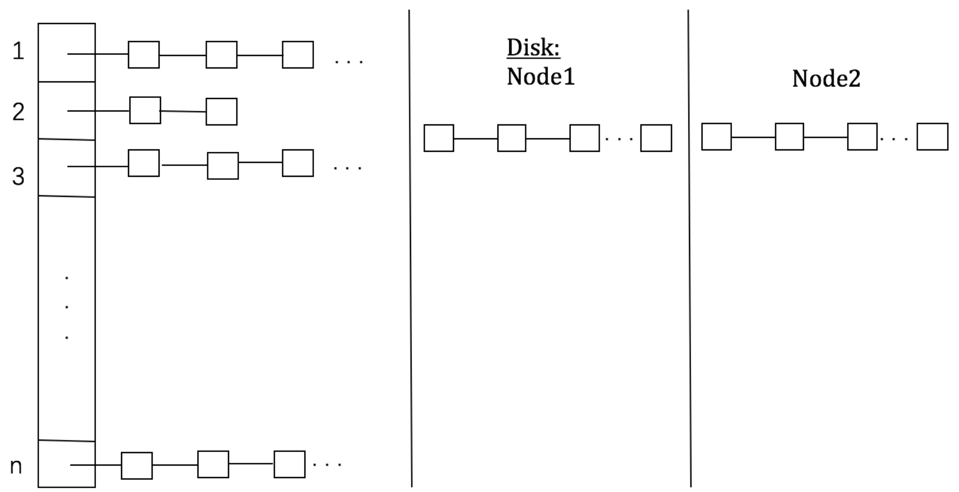
insert (u, NEAR[u]) into T;

for every w ∈ Adj(u) do *Adj(u) → Adjacent to u

if weight(w, NEAR[w]) > weight(w,u) then

NEAR[w] = u;

- Assume that the input is in adjacency lists form



- Assume that $M = \theta(n)$, n = |V|. This means that Q can be stored in core memory;

Analysis:

- Step1: takes $\frac{|E|}{B}$ I/O operations
- Step2: takes $\leq \frac{2|V|}{B}$ I/O operations
- Step3: No I/O operations; Let d_u be the degree of u for any $u \in V$.
- Step4: takes $\leq \sum_{u \in V} \left\lceil \frac{d_u}{B} \right\rceil \leq \sum_{u \in V} \left(\frac{d_u}{B} + 1 \right) = O\left(\frac{|E|}{B} + |V| \right)$ I/O operations.

Note that this algorithm is optimal when $|E| \geq |V|B$.