M22AI608

Kadapala Rakesh Reddy

02-July-23

ALGORITHMS FOR BIGDATA - CSL7630

MINOR - II

IIT-J - D.C.S - M.Tech - Summer sem-I

(1)

(1) False:- streaming algorithms process data in a memory-efficient manner, often by utilizing limited portions of input at a time.

(11) False:- streaming algorithms can be used & Handles for both Real time data streams & static datasets.

(111) True:- streaming algorithms can't Handle Parallel Processing of data

(IV) False:- S.A often Provide approximate results for computations on streaming data

(V) False:- universal Hashing doesn't mandate the use of cryptographic hash functions

(VI) False:- universal hasing minimizes collisions but doesn't guarantee unique locations for all keys

(VII) False:- universal Hashing doesn't guarantee constant time complexity for Hash Table operations

(VIII) False:- Property Testing applies to both Quantitative & Qualitative properties

(IX) True:- Property Testing algorithms can Handle dynamic datasets that change overtime

(X) False:- Property testing aims to provide Probabilistic results in sublinear time, not exact solutions

(2)

Item	2	3	1	4	2	3	1	3	1	4	5	3	2	1	6
count	1	1	1	1	2	2	2	3	3	2	1	4	3	4	1

Here algorithm initializes a table with a row for each possible item in stream & column for count of no. of times that item has been seen

The Algorithm then iterates through data stream, updating the counts for each item as it goes. If count for an item reaches k, then item is added to output set. If count for an item falls below k, then item is evicted from output set.

The Algorithm terminates when the end of data stream is reached

The final o/p of algorithm is set of items with a count at least k

So final o/p of set = {1, 2, 3, 4, 5}

The misra-Gries algorithm has several benefits over Traditional frequency counting approaches for large streams of data. first it is very memory efficient

- Algo. only needs to store a small Table with a row for each possible item in stream.
- This is contrast to Traditional frequency counting approaches, which may need to store count for each item in stream.

Reservoir sampling :- Initialize reservoir size s

- for each item in the data stream
- if item not in reservoir
- remove a random item from a reservoir
- final o/p is set of items in reservoir.

final o/p = {2, 3, 1, 4, 6}

Reservoir sampling is a simple & efficient Algorithm for randomly selecting a subset of size 'k' from a large stream of data.

The Algo. works by maintaining a reservoir of size 'k'.

- As each item is added to reservoir with probability $\frac{1}{k}$ if the reservoir is full, random item is removed to make a new item.

(3) (i) A universal one-way Hash function family is a set of Hash functions such as that for any 2 distinct keys, the probability that they hash to same value is at most $1/m$ where m = size of Hash Table.

(ii) A solution that utilizes a universal hash functions family to speed up searching a keywords across a large set of text documents could be a hash table. The Hash would be initialized with set of Hash functions from universal Hash function family. when a keyword is searched for, it would be Hashed using each of Hash functions in family. The results of Hash functions would then be used to lookup the keywords in HashTable. If the keyword found in HashTable, Then it is returned, else keyword is not found.

(iii) The properties of this soln are that it requires a large hash table. This is because hash table must be at least large as set of keywords and designed to minimize the probability of collisions. This means the search operation can be performed quickly even for large set of keywords.

This soln also effective because it can find any keyword in set even if if keyword is not stored in Hash Table.

The Trade offs of this soln are that it requires a large Hash Tables. This is because HashTable must be at least as large as set of keywords..

However Probability of this happening is very small as long as Hash functions from universal Hash functions family are used.

(iv) A universal Hash function family was req. in soln because it ensures that probability of collisions is minimized

This is imp. because it ensures probability of collisions can significantly can slowdown the search operation. if a non universal Hash function used, the probability of collisions could be High especially for large set of keywords.

It is not possible to use any other Hash function in this method without increasing probability of collisions. This is because the universal hash function family is designed to minimize the probability of collisions.

Any other Hash functions would likely have a Higher probability of collisions, which could make search operation slower.

(4)

A graph is connected if There is a Path b/w any 2 vertices in graph. A connected graph does not consist of disconnected components; every vertex is reachable from every other vertex.

G=(V, E) Here every Pair of vertices u & v in V. There exists a Path b/w u & v.

Here consecutive pair of vertices is connected with edge E.

necessary conditions:

(a) graph must have atleast 2 vertices
(b) must be atleast one edge in graph

(ii) Process & steps

1) choose random subset of vertices in graph
2) Run algorithm on subset of vertices
3) if algo. o/p That The graph is connected, graph is considered connected else not connected.

(iii) Ex

$$\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

choosing random subset of vertices {0, 1, 2}

- if Algorithm graph is connected graph considered as connected else not connected.
- if it is connected because of there is a path b/w subset of vertices.

∴ So Algo. collectly determins connectedness of graph