大数据计算基础

第二章 大数据算法

哈尔滨工业大学
刘显敏
liuxianmin@hit.edu.cn

大数据计算基础

第二章 大数据算法——流算法（亚线性算法）

哈尔滨工业大学
刘显敏
liuxianmin@hit.edu.cn

本讲内容

- 亚线性算法的定义
- 水库抽样—空间亚线性算法
- 平面图直径—时间亚线性计算算法
- 全0数组判定—时间亚线性判定算法
- 数据流中频繁元素
- 最小生成树
- 序列有序的判定

亚线性的含义

- 时间/空间/IO/通讯/能量等消耗是**o(输入规模)**
- 亚线性时间算法
  - 亚线性时间近似算法
  - 性质检测算法
- 亚线性空间算法
  - 数据流算法

亚线性时间问题

- 给定一个社交网络，如何平均每个人的朋友个数，即在大图中计算其结点的平均度
- 能否在不访问所有顶点的情况下完成此任务？
  - 精确计算需要访问最少n-1个顶点
  - 是否可以简单的抽样？

亚线性空间问题

…….1, 3, 23, 3, 34, 23, 41

有限内存

中位数

- 一个（源源不断到来的）数据集合（流），只能扫描一次，如何求其中位数？
  - 不能存储所有数据->不能对其进行排序
  - 应当存储哪些数据？

本讲内容

- 亚线性算法的定义
- **水库抽样—空间亚线性算法**
- 平面图直径—时间亚线性计算算法
- 全0数组判定—时间亚线性判定算法
- 数据流中频繁元素
- 最小生成树
- 序列有序的判定

水库抽样——一个亚线性空间算法

- **输入：一组数据，其大小未知**
- **输出：这组数据的k个均匀抽样**
- **要求：**
  - 仅扫描数据一次
  - 空间复杂性为O(k)
  - 扫描到数据的前n个数字时(n>k)，保存当前已扫描数据的k个均匀抽样

水库抽样算法

1. 申请一个长度为*k*的数组*A*保存抽样
2. 保存首先接收到的*k*个元素
3. 当接收到第*i*个新元素*t*时，以*k*/*i*的概率随机替换*A*中的元素(即生成[1,*i*]间随机数*j*, 若*j*≤*k*, 则以*t*替换*A*[*j*])

性质1：该采样是均匀的

$$\frac{k}{i}\times\left(1-\frac{1}{i+1}\right)\times\left(1-\frac{1}{i+2}\right)\times\cdots\times\left(1-\frac{1}{n}\right)=\frac{k}{n}$$

性质2: 空间复杂性是O(*k*)

本讲内容

- 亚线性算法的定义
- 水库抽样—空间亚线性算法
- 平面图直径—时间亚线性计算算法
- 全0数组判定—时间亚线性判定算法
- 数据流中频繁元素
- 最小生成树
- 序列有序的判定

平面图的直径——一个亚线性时间计算算法

- **输入：** ***m*个顶点的平面图，任意两点之间的距离存储在矩阵*D*中，即点*i*到点*j*的距离为D_{ij}**
  - 输入大小是$n=m^2$
  - 最大的D_{ij}是图的直径
  - 点之间的距离对称且满足三角不等式
- **输出：该图的直径和距离最大的D_{ij}**
- **要求：**
  - 运行时间为$o(n)$

平面图的直径近似算法

- **无法在要求的时间内得到精确解，寻找近似算法**
- **近似算法**

1. 任意选择$k \le m$
2. 选择使得D_{kl}最大的l
3. 输出D_{kl}和(k, l)

- **近似比**

$$D_{ij} \le D_{ik} + D_{kj} \le D_{kl} + D_{kl} \le 2\, D_{kl}$$

因而近似比为2

- **运行时间**

$$\mathbf{O}(\boldsymbol{m})=\mathbf{O}(\sqrt{\boldsymbol{n}}) = \boldsymbol{o}(\boldsymbol{n})$$

近似算法

- **什么是近似算法**
  - 近似算法主要用来解决优化问题
  - 能够给出一个优化问题的近似优化解的算法
- **近似算法解的近似度**
  - 问题的每一个可能的解都具有一个代价
  - 问题的优化解可能具有最大或最小代价
  - 我们希望寻找问题的一个误差最小的近似优化解
- **我们需要分析近似解代价与优化解代价的差距**
  - Ratio Bound
  - 相对误差

近似比

- **Ratio Bound**

设A是一个优化问题的近似算法, A具有ratio bound $p(n)$, 如果

$$\max\left\{\frac{C}{C^*},\frac{C^*}{C}\right\}\le p(n)$$

其中n是输入大小, C是A产生的解的代价, C^*是优化解的代价.

- 如果问题是最大化问题, max$\{C/C^*, C^*/C\}=C^*/C$
- 如果问题是最小化问题, max$\{C/C^*, C^*/C\}=C/C^*$
- 由于$C/C^*<1$当且仅当$C^*/C>1$, Ratio Bound不会小于1
- Ratio Bound越大, 近似解越坏

- **相对误差**

相对误差: 对于任意输入, 近似算法的相对误差定义为$|C-C^*|/C^*$, 其中C是近似解的代价, C^*是优化解的代价.

相对误差界: 一个近似算法的相对误差界为$\varepsilon(n)$, 如果$|C-C^*|/C^*\le\varepsilon(n)$.

本讲内容

- 亚线性算法的定义
- 水库抽样—空间亚线性算法
- 平面图直径—时间亚线性计算算法
- **全0数组判定—时间亚线性判定算法**
- 数据流中频繁元素
- 最小生成树
- 序列有序的判定

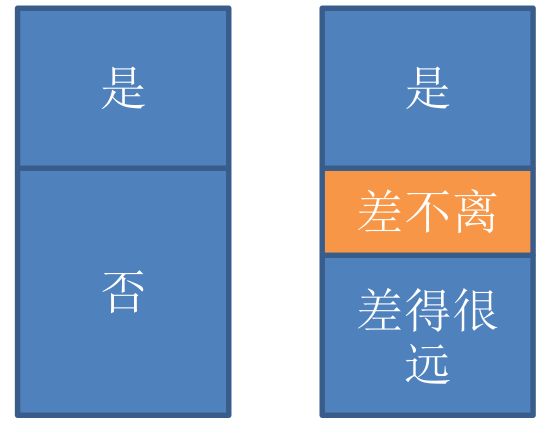
全0数组的判定——一个亚线性时间判定算法

- **输入：包含n个元素的0,1数组A**
- **输出：A中的元素是否全是0**
- **要求：**
  - 运行时间为$o(n)$

判定问题的近似

- 无法在要求的时间内得到精确解，寻找近似解
  - 判定问题如何近似？
- 输入满足某种性质或者**远**非满足此性质

问题：图片中是否包含“猫”



- ε-远离
  - 对于输入x，如果着从x到L中任意字符串的汉明距离至少为 $\varepsilon|x|$，则x是ε-远离 L的.
- 全0数组判定问题的近似
  - **是否A=00…0或者其包含1的个数大于εn?**

全0数组的判定近似算法

- **算法描述**

1. 在A中随机独立抽取$s=2/\varepsilon$个位置上的元素
2. 检查抽样，若不包含1，则输出“是”，若包含1，则输出“否”

- **判定精确性分析**
  - **如果A是全0数组，始终输出”是”**
  - **如果A是ε-远离的，$\Pr[\text{error}]=\Pr[\text{抽样中没有1}]\le(1-\varepsilon)^s\approx e^{-\varepsilon s}=e^{-2}<\frac{1}{3}$**

- **运行时间：**$O(s)$

- **证据引理**

如果一次测试以大于等于p的概率获得一个证据，那么s=2/p轮测试得到证据的概率大于等于2/3

判定算法的定义

- **对于判定问题L，其查询复杂性为q(n)和近似参数ε的性质测试算法是一个随机算法，其满足对于给定L的是一个实例x, 最多进行q(|x|)次查询，并且满足下述性质：**
  - **如果x在L之中，该算法以最少2/3的概率返回“是”**
  - **如果x 是ε远离L的, 该算法以最小2/3的概率返回“否”**

本讲内容

- 亚线性算法的定义
- 水库抽样—空间亚线性算法
- 平面图直径—时间亚线性计算算法
- 全0数组判定—时间亚线性判定算法
- **数据流中频繁元素**
- 最小生成树
- 序列有序的判定

大数据的数据流模型

- 数据只能顺序扫描1次或几次
- 能够使用的内存是有限的
- 希望通过维护一个内存结果(概要)来给出相关性质的一个有效估计
- 数据流模型适用于大数据
  - **顺序扫描数据仅一次**
  - **内存亚线性**

数据流模型

- 来自某个域中的元素序列

  $< x_1, x_2, x_3, x_4, \ldots.. >$
- 有限的内存:

  内存<< 数据的规模

  通常 $O(\log^k n)$ 或 $O(n^\alpha)$ for $\alpha < 1$
- 快速处理每个元素

从数据流中计算什么？

32，112，14，9，37，83，115，2，

容易计算的函数: min, max, sum, …

使用单个寄存器 s, 直接更新:

- max: 初始化 $s \leftarrow 0$

  对于元素 x , $s \leftarrow \max\{s, x\}$

- sum: 初始化 $s \leftarrow 0$

  对于元素 x , $s \leftarrow s + x$

  - “概要”是单个值
  - 是可合并的

频繁元素

32, 12, 14, 32, 7, 12, 32, 7, 6, 12, 4,

- 元素出现多次，希望找到出现最频繁的元素
- n：不同元素的数量
- m：数据流中元素个数

频繁元素

32, 12, 14, 32, 7, 12, 32, 7, 6, 12, 4,

应用:

- 网络: 找到“elephant flow”
- 搜索: 找到频繁查询

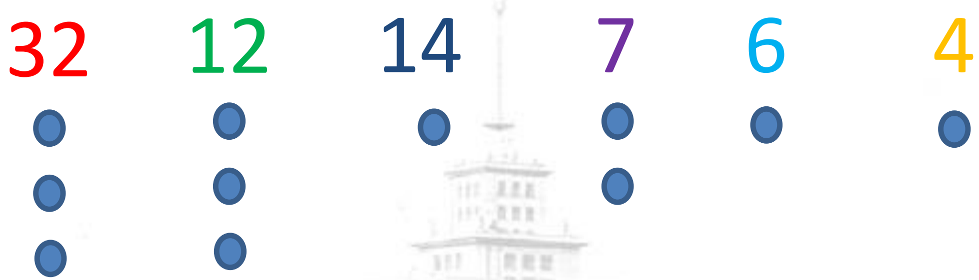
Zipf原则: 典型的频率分布是高度偏斜的，只有少数频繁元素.
最多10%的元素占元素总个数的 90%.
我们发现出现次数最多的元素

频繁元素: 精确解

32, 12, 14, 32, 7, 12, 32, 7, 6, 12, 4,

精确解:

- 对每一个单独元素设置一个计数器
- 当处理一个元素时，增加相应计数器



问题: 需要维护n个计数器

但只能有$k \ll n$ 个计数器

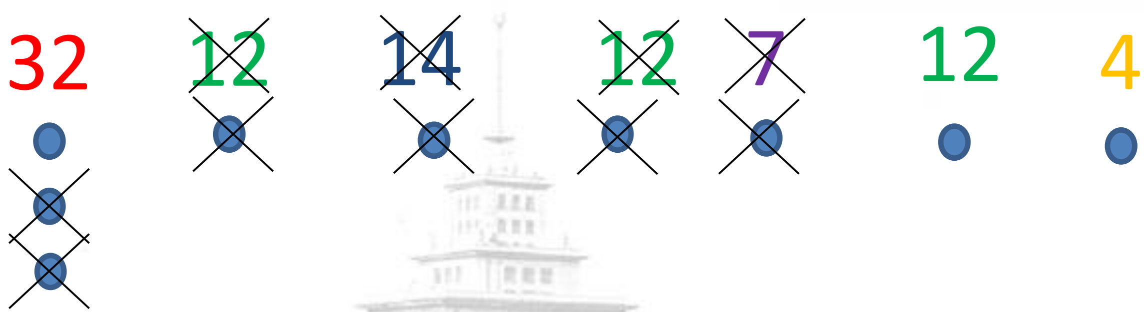
频繁元素计算算法

Misra Gries(MG)算法

32, 12, 14, 32, 7, 12, 32, 7, 6, 12, 4,

<u>处理元素x</u>

- If 已经为x分配计数器,增加之
- Else If 没有相应计数器，但计数器个数少于k,为x分配计数器，并设为**1**.
- Else, 所有计数器减**1**.删除值为**0**的计数器.



$$n = 6$$
$$k = 3$$
$$m = 11$$

频繁元素算法

32, 12, 14, 32, 7, 12, 32, 7, 6, 12, 4,

处理元素x

- If 已经为x分配计数器,增加之
- Else If 没有相应计数器，但计数器个数少于k,为x分配计数器，并设为$\mathbf{1}$.
- Else, 所有计数器减$\mathbf{1}$.删除值为$\mathbf{0}$的计数器.

x 出现几次?

- If 我们有一个x的计数器, 返回其值
- Else, 返回$\mathbf{0}$.

该估计显然过低
如何精确估计?

分析

一个计数器x减少了几次?

$\Leftrightarrow$ 我们有几个减少计数器的步骤?

- 整个结构的权重(计数器的和)记作m'
- 整个数据流的权重(全部元素的数量)是m
- 每一个计数器降低的步骤减少k个计数，但是并未计入输入元素的此次出现，即$k+1$次未计入的元素出现.

$\Rightarrow$ 最多有$\frac{m-m'}{k+1}$个减少步骤

$\Rightarrow$ **估计值和真实值相差最多 $\frac{m-m'}{k+1}$**

分析

估计值与真实值相差最多 $\boldsymbol{\frac{m-m'}{k+1}}$

$\Rightarrow$当数据流中元素的总数$\gg \frac{m-m'}{k+1}$时，**得到**x的一个好的估计

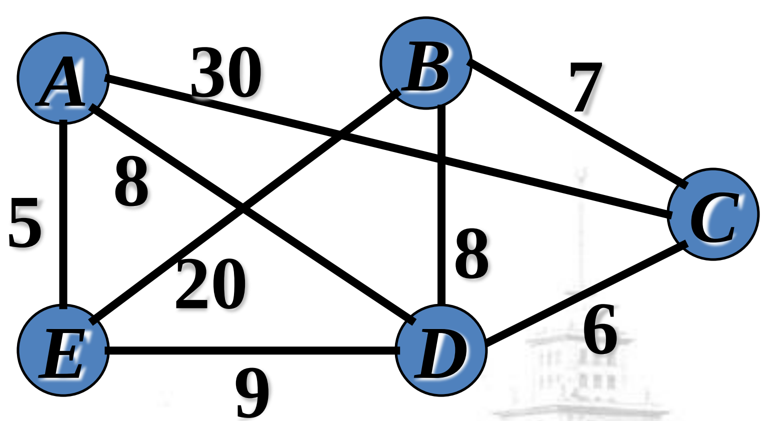
- 错误的界限和k成反比
- 利用概要计算错误的界限：记录m，计算m'和k.
- 该算法有效的原因：“Zipf原则”

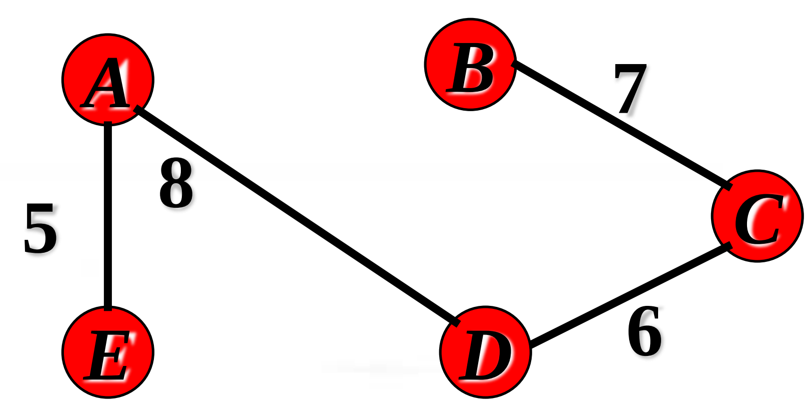
本讲内容

- 亚线性算法的定义
- 水库抽样—空间亚线性算法
- 平面图直径—时间亚线性计算算法
- 全0数组判定—时间亚线性判定算法
- 数据流中频繁元素
- **最小生成树**
- 序列有序的判定

问题的定义

- **输入：** 无向有权连通图G=(V, E)，其顶点的度最大为D,边上的权来自整数集合{1, ..., W}
- **输出：** 图G的最小生成树的权重





精确解

- 贪心法
  - Prime算法
  - Kruskal算法
- 时间复杂性：O($m$log$n$)
- 超过线性

亚线性算法的假设

- 图组织成邻接表的形式
  - 可以直接访问每个结点的邻居
- 可以随机均匀地选择结点

时间亚线性算法的思想

- 利用特定子图连通分量的数量估计最小生成树的权重
- 假设所有边的权重都是1或者2，最小生成树的权重

  $=\#N_1+\#N_2$ ($\#N_i$: 最小生成树中权重至少为i的边的数量)

  $=n-1+\#N_2$ (最小生成树有n-1条边)

  $=n-1+$权重为1边构成的导出子图的连通分量数-1

最小生成树和连通分量的关系

- 一般的情况
  - G_i: G中包含所有权重小于i的边的子图
  - C_i: G_i中的连通分量数
  - 最小生成树权重大于i的边数为C_i-1
- $W_{MST}(G) = n - w + \sum_{i=1}^{w-1} C_i$

证明：

令β_i为最小生成树中权重大于i的边的个数

每一条MST边对WMST基础贡献为1，每个权重大于1的边额外贡献了1，每条权重大于2的边贡献的更多，因此

$$W_{MST}(G) = \sum_{i=0}^{w-1} \beta_i = \sum_{i=0}^{w-1} (C_i - 1) = -w + \sum_{i=0}^{w-1} C_i = n - w + \sum_{i=1}^{w-1} C_i$$

基础算法：连通分量个数的估计

- 输入：图$G=(V, G)$，有n个顶点，表示为邻接矩阵，结点最大度为d
- 输出：连通分量的个数
- 精确解时间复杂性：$\Omega(dn)$
- 利用随机化方法
- 估计连通分量个数#CC
  - #CC$\pm\varepsilon n$的概率$\geq 2/3$
  - 运行时间和n无关

估计连通分量的方法：核心思想

- C:连通分量的个数
- 对于每个结点u, n_u: u所在连通分量的结点数
- 对于每个连通分量：$\sum_{u \in A} \frac{1}{n_u} = 1$，
- 故: $\sum_{u \in V} \frac{1}{n_u} = C$
- 通过估计抽样顶点的n_u来估计这个和
  - 如果u所在的分量很小，其规模可以通过BFS估计
  - 如果u所在连通分量很大，$1/n_u$很小，对和的贡献很小
  - 可以在几步以内完成BFS

每个u所在连通分量结点数的估计

- 令$\hat{n}_u$=min{n_u, 2/ε}
  - 当结点数小于2/ε时，$\hat{n}_u = n_u$
  - 否则，$\hat{n}_u$= 2/ε，因此$0 < \frac{1}{\hat{n}_u} - \frac{1}{n_u} < \frac{1}{\hat{n}_u} = \frac{\varepsilon}{2}$
- 在这种情况下，对C的估计

$$\hat{C} = \sum \frac{1}{\hat{n}_u}$$

- 则$\left|\hat{C} - C\right| = \left|\sum\left(\frac{1}{\hat{n}_u} - \frac{1}{n_u}\right)\right| \leq \frac{\varepsilon n}{2}$

连通分量数估计算法

CC(G, d, ε)

1. for i=1 to $s=\theta(\frac{1}{\varepsilon^2})$ do
2. 随机选择点u
3. 从u开始BFS，将访问到的顶点存到排序序列L中，访问完连通分量或L=2/ε时停止，$\hat{n}_u = |L|$
4. N=N+$1/\hat{n}_u$
5. 返回$\tilde{C} = N/s \times n$

运行时间：$O(\frac{d}{\varepsilon^3}\log\frac{1}{\varepsilon})$

连通分量近似计数的分析

- 分析的目的：$\Pr[|\tilde{C}-\hat{C}|>\frac{\varepsilon n}{2}]\le\frac{1}{3}$
  - 估计值和真实值相差过大的概率很小
- 对于采样中的第i个结点u, 令$Y_i = {}^{1}/_{\hat{n}_u}$
- $Y=\sum_{i=1}^{s}Y_i=\frac{s\tilde{C}}{n}$
- $\mathrm{E}[Y]=\sum_{i=1}^{s}E[Y_i]=sE[Y_1]=s\cdot\frac{1}{n}\sum_{u\in V}\frac{1}{\hat{n}_v}=\frac{s\hat{C}}{n}$

$\Pr\left[|\tilde{C}-\hat{C}|>\frac{\varepsilon n}{2}\right]=Pr[|\frac{n}{s}Y-\frac{n}{s}E[Y]|>\frac{\varepsilon n}{2}]=Pr[|Y-E[Y]|>\frac{\varepsilon s}{2}]$

连通分量近似计数的分析(续)

Hoeffding界

Y1,...Ys为[0,1]区间内独立同分布的随机变量，令 $Y=\sum_{i=1}^{s} Y_i$，则$Pr[|Y-E[Y]|\geq \delta]\leq 2e^{-2\delta^2/s}$

$$\Pr\left[\left|\tilde{C}-\hat{C}\right|>\frac{\varepsilon n}{2}\right]=Pr\left[|Y-E[Y]|>\frac{\varepsilon s}{2}\right]$$

$$\leq 2e^{-\frac{\varepsilon^2 s}{2}}$$

$$s=\theta\left(\frac{1}{\varepsilon^2}\right)\Rightarrow \Pr\left[\left|\tilde{C}-\hat{C}\right|>\frac{\varepsilon n}{2}\right]\leq\frac{1}{3}$$

连通分量近似技术的分析(续)

$$\Pr[|\tilde{C}-\hat{C}|>\frac{\varepsilon n}{2}]\leq\frac{1}{3}$$

$$|\hat{C}-C|\leq\frac{\varepsilon n}{2}$$

因此，下列事件发生的概率大于2/3:

$$|\tilde{C}-C|\leq|\tilde{C}-\hat{C}|+|\hat{C}-C|\leq\frac{\varepsilon n}{2}+\frac{\varepsilon n}{2}=\varepsilon n$$

综上所述，有n个顶点的图中，若其顶点的度至多为d，则其连通分量的数量估计误差最多为$\pm\varepsilon n$

最小生成树近似算法

1. for i=1 to w-1 do
2. $\tilde{C}_i = CC\left(G_i, d, \frac{\varepsilon}{w}\right)$
3. return $\tilde{w}_{MST} = n - w + \sum_{i=1}^{w-1} \tilde{C}_i$

并集界限

对于事件$A_1, \ldots, A_n$

$$P(\bigcup_i A_i) \le \sum_i P(A_i)$$

分析：

- 假设C_i的所有估计都是正确的，$|\tilde{C}_i - C_i| \le \frac{\varepsilon}{w} n$，则$|\tilde{w}_{MST} - w_{MST}| = |\sum_{i=1}^{w-1}(\tilde{C}_i - C_i)| \le \sum_{i=1}^{w-1}\left|\tilde{C}_i - C_i\right| \le w \cdot \frac{\varepsilon}{w} n = \varepsilon n$
- Pr[所有w-1次估计都正确]≥(2/3)$^{w-1}$
- 这并不够好! 可以通过CC算法中,取合适的s值，使得每一轮的错误概率≤$\frac{1}{3w}$，应当取多少呢?
- 利用并集界限，Pr[error]≤ $w \cdot \frac{1}{3w} = \frac{1}{3}$

乘近似

- 对于MST的代价，可以从加的近似导出乘的近似
  - $w_{\mathrm{MST}} \geq n\text{-}1 \Rightarrow w_{\mathrm{MST}} \geq n/2\ (\mathrm{n} \geq 2)$
- εn加近似
  - $w_{MST} - \varepsilon n \leq \widehat{w}_{MST} \leq w_{MST} + \varepsilon n$
- $(1 \pm 2\varepsilon)$乘近似
  - $w_{MST}(1 - 2\varepsilon) \leq w_{MST} - \varepsilon n \leq \widehat{w}_{MST} \leq w_{MST} + \varepsilon n \leq w_{MST}(1 + 2\varepsilon)$

本讲内容

- 亚线性算法的定义
- 水库抽样—空间亚线性算法
- 平面图直径—时间亚线性计算算法
- 全0数组判定—时间亚线性判定算法
- 数据流中频繁元素
- 最小生成树
- 序列有序的判定

数组有序性判定

- 输入：n个数的数组, $x_1, x_2, \ldots, x_n$
- 输出：这个数组是否有序？
  - 需要访问这n个数，时间是$\Omega(n)$
- 近似版本
  - 这个数组是有序的还是ε远离有序的？
- ε远离
  - 我们必须删除大于εn个元素才能保证剩下的元素有序

亚线性算法

for k=1 to $2/\varepsilon$ **do**

  选择数组中第i个元素x_i

  用x_i在数组中做二分查找

  if 发现$i<j$ 但是$x_i>x_j$ then //碰到了“坏”索引

    return false

return true

算法的时间复杂性：$O(\frac{1}{\varepsilon}\log n)$

算法精确性

- 输入数列有序，则总返回True
- 下面证明：当输入数列ε远离有序时，算法返回false的概率大于2/3
- **证据引理：**如果一次测试以大于等于p的概率获得一个证据，那么$s=2/p$轮测试得到证据的概率大于等于2/3

首先证明：如果输入ε远离有序，则存在大于

εn个“坏”索引

性质的证明

- **如果输入ε远离“有序”，则存在大于εn个“坏”索引**

证明：我们证明其逆否命题，即如果“坏”索引的个数小于εn, 则其存在一个长度大于εn的单调递增子序列.

对于任意"好"索引i和j，$x_i<x_j$

令k是在二分搜索中将x_i和x_j分开的最近顶点，也就是对于整个数组建一个二分搜索树，在二分搜索树中x_i和x_j的最近公共祖先，则$i<k<j$，因为i和j都是“好”索引，那么$x_i<x_k<x_j$

性质的证明(续)

- 当输入数列ε远离有序时，算法返回false的概率大于2/3

证明:往证算法返回true的概率小于1/3

我们已经证明，如果输入ε远离有序，则存在大于εn个“坏”索引，即数组中“坏”索引的概率大于$\boldsymbol{\varepsilon}$。

当数组中“坏”索引的概率大于$\boldsymbol{\varepsilon}$时，选择的索引都是好的概率小于$(1-\varepsilon)^{2/\varepsilon} < e^{-2} < \frac{1}{3}$

总结