

有序序列搜索问题最快算法为二分法的一个理论证明

刘耕滔

浙江师范大学, 数学与计算机科学学院, 浙江 金华
Email: 001lgt@zjnu.edu.cn

收稿日期: 2021年2月16日; 录用日期: 2021年3月16日; 发布日期: 2021年3月22日

摘要

为了证明有序序列搜索问题最快算法为二分法, 先由一个具体例子引入, 得到问题描述与三条初步结论。再结合二分法定义由计算均值方法得到评价算法平均收敛速度的标准。最后由数学归纳法证明有序序列搜索问题每次迭代中迭代数的性质, 进而证明解决有序序列搜索问题的所有算法中, 二分法算法可以达到平均最快的收敛速度。

关键词

二分法, 有序序列, 搜索问题, 均值

A Theoretical Proof of Dichotomy as the Fastest Algorithm for Sequential Search Problem

Gengtao Liu

College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua Zhejiang
Email: 001lgt@zjnu.edu.cn

Received: Feb. 16th, 2021; accepted: Mar. 16th, 2021; published: Mar. 22nd, 2021

Abstract

In order to prove that the fastest algorithm for the ordered sequence search problem is method of bisection, a specific example is introduced, and the problem description and three preliminary

conclusions are obtained. Combined with the definition of dichotomy, the standard of evaluating the average convergence rate of the algorithm is obtained by calculating the mean value method. Finally, the properties of the iterated algebra in each iteration of the ordered sequence search problem are proved by mathematical induction, and it is proved that the dichotomy algorithm can achieve the fastest average convergence speed among all the algorithms for solving the ordered sequence search problem.

Keywords

Method of Bisection, Ordered Sequence, Search Problem, Mean Value

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

众所周知, 对于求根类的问题, 二分法是最快的算法。如解决以下问题:

甲和乙在进行一项游戏, 甲随机记下某个在 1 至 1000 内的自然数让乙来猜, 乙依据某种固定的选取整数的方式来猜数, 乙每猜一个数, 甲就会告诉乙这个数是否为甲记下的数, 若不是, 则会告诉乙他的猜测是偏大还是偏小, 那么乙如何选取整数可以最快地猜到甲记下的数字?

2. 问题描述

将上述问题的一般形式转化为一个数学问题:

从小到大连续取 n (n 不为 0 或无穷) 个整数 a_1, a_2, \dots, a_n , 组成集合 $A = \{a_1, a_2, \dots, a_n\}$, 等概率地随机抽取一个数 $a_\eta \in A$ ($\eta \in \mathbb{N}_+, 1 \leq \eta \leq n$)。

取 $a_{\xi_1} \in A$ ($\xi_1 \in \mathbb{N}_+, 1 \leq \xi_1 \leq n$) 将集合 A 分为两个集合:

$$A_{u1} = \{m \mid \xi_1 < m \leq n, m \in \mathbb{N}_+\} \text{ 与 } A_{d1} = \{m \mid 1 \leq m < \xi_1, m \in \mathbb{N}_+\}$$

判断 $a_{\xi_1} = a_\eta$ 是否成立, 若不成立, 则判断 $a_{\xi_1} < a_\eta$ 是否成立。

若 $a_{\xi_1} < a_\eta$ 成立, 则取 $a_{\xi_2} \in A_{u1}$ ($\xi_2 \in \mathbb{N}_+, \xi_1 < \xi_2 \leq n$) 将集合 A_{u1} 分为两个集合:

$$A_{u2} = \{m \mid \xi_2 < m \leq n, m \in \mathbb{N}_+\} \text{ 与 } A_{d2} = \{m \mid \xi_1 < m < \xi_2, m \in \mathbb{N}_+\}。$$

若 $a_{\xi_1} < a_\eta$ 不成立, 则取 $a_{\xi_2} \in A_{d1}$ ($\xi_2 \in \mathbb{N}_+, 1 \leq \xi_2 < \xi_1$) 将集合 A_{d1} 分为两个集合:

$$A'_{u2} = \{m \mid \xi_2 < m < \xi_1, m \in \mathbb{N}_+\} \text{ 与 } A'_{d2} = \{m \mid 1 \leq m < \xi_2, m \in \mathbb{N}_+\}。$$

依此类推, 重复以上取 a_{ξ_i} ($i=1, 2, \dots$) 以及判断的步骤。最终一定可以在有限步后取得 $a_{\xi_i} = a_\eta$ ($i \in \mathbb{N}_+$)。

问: 运用什么算法选取 a_{ξ_i} ($i=1, 2, \dots$), 算法的平均收敛速度最快?

3. 初步分析

根据题目, 实际上对于有序序列的一个搜索问题, 可以初步分析得到以下结论:

- 1) 根据算法的性质 [1], 当选取 a_{ξ_i} ($i=1, 2, \dots$) 的方式与集合 A 元素个数 n 确定时, 对于任意取定的 $a_\eta \in A$ ($\eta \in \mathbb{N}_+, 1 \leq \eta \leq n$), 最终取得 $a_{\xi_i} = a_\eta$ ($i \in \mathbb{N}_+$) 时所迭代的平均次数 i 唯一确定。
- 2) 最终取得 $a_{\xi_i} = a_\eta$ ($i \in \mathbb{N}_+$) 时对应的迭代数 i 不大于集合 A 元素个数 n 。

3) 每当在 A 的子集中选取 a_{ξ_i} ($i=1,2,\dots$) 后, 该子集必可以依据 a_{ξ_i} ($i=1,2,\dots$) 分为两个真子集(包含空集)。

下证解决本问题的所有算法中, 二分法算法可以达到平均最快的收敛速度; 首先给出针对本题二分法的定义与证明所需引理。

定义 1 二分法算法步骤[2]:

Step1: 取 $\xi = \left\lceil \frac{n}{2} \right\rceil$, $a_{\xi} \in A$ 。若 $a_{\xi} = a_n$, 则算法结束。

Step2: 若 $a_{\xi} < a_n$, 则置集合 A_u 为更新的集合 A ; 若 $a_{\xi} > a_n$, 则置集合 A_d 为更新的集合 A 。

Step3: 更新 n 为集合 A 的元素个数, 返回 Step1。

比较针对本问题的各种算法的收敛速度时, 由于取定任意 $a_n \in A$ 的概率相同, 从而可以比较取得任意取定的 $a_n \in A$ 的迭代数的均值, 均值越小的算法收敛速度越快。该均值直接从正面求解与证明较为困难, 而取定任意 $a_n \in A$ 的概率相同, 且根据集合与算法的确定性, 对于某些算法当集合 A 的基数 n 与选取元素的算法确定时, 每一个元素所对应的取得该元素时算法的迭代数唯一确定, 因此可将解决问题思路转化为: 先根据算法确定好每个元素的各项参数, 再随机选取一个元素作为 a_n 的方式计算迭代数的均值。转化后的问题与原问题等价。

设元素 a_i 对应的迭代数为 w_i , 则计算均值的公式[3]:

$$E = \sum_i \left(\frac{1}{n} \cdot w_i \right) = \frac{1}{n} \cdot \sum_i w_i$$

根据公式, 在问题规模相同的情况下只需比较每个元素对应的迭代数和 $\sum_i w_i$, 该和数越小则对应算法收敛速度越快。因此, 可将元素与根据某算法取得该元素的迭代数列成表格, 再计算表中所有迭代数的均值以比较各算法的收敛速度。

4. 两个命题

命题 1 对于有序序列搜索问题, 任何算法下全体元素所对应的迭代数中 p ($p=1,2,\dots$) 至多出现 2^{p-1} 次。

证明: 使用数学归纳法证明。

令 $p=1$ 时, 显然成立。

设 $p=k$ ($k=2,3,\dots$) 时成立, 且根据结论(3), 算法在选取对应的元素后, 剩余可行的元素集合被算法迭代选取的元素至多分成两部分, 则当 $p=k+1$ 时, 根据算法可以在初始集合 $A = \{a_1, a_2, \dots, a_n\}$ 的至多 $2 \cdot 2^{k-1}$ 个互不相交的真子集中的一个选取下一个元素, 因此至多有 2^k 个元素对应迭代数为 $k+1$, 根据数学归纳法, 原命题成立。

命题 2 解决有序序列搜索问题的所有算法中, 二分法算法可以达到最快的平均收敛速度。

证明: 根据二分法, 每次取得可行区间中点的近似的元素。设 $2^{i-1} \leq n < 2^i$ ($i \in \mathbb{N}_+$), 则第 p ($p=1, \dots, i-1$) 次存在 2^{p-1} 个可取元素, 第 i 次存在 $n - 2^{i-1} + 1$ 个可取元素。

根据上文列出二分法算法的元素与迭代数的对应表, 再根据命题 1, 二分法算法使得小于 i 的迭代数对应的元素数达到了最大值, 其余元素均为第 i 次迭代时所取得, 不存在大于 i 的迭代数对应的元素, 因此迭代数之和 $\sum_i w_i$ 达到了最小值, 命题 2 成立。

注: 二分法算法可以达到最快的平均收敛速度, 但可以达到最快的平均收敛速度的算法并不唯一, 对于不同的具体问题, 只要可以满足命题 1 中的最多条件, 即可达到最快的收敛速度。

参考文献

- [1] 郑莉, 董渊, 编著. C++程序设计基础教程[M]. 北京: 清华大学出版社, 2010.
- [2] 严蔚敏, 吴伟民, 编著. 数据结构 C 语言版[M]. 北京: 清华大学出版社, 1997: 13.
- [3] 茆诗松, 程依明, 濮晓龙, 编著. 概率论与数理统计教程[M]. 北京: 高等教育出版社, 2019.