

20220428-书&数据结构

1.过程描述

1.1数据结构与算法

两个值得探究的问题

计算

计算模型

2.结果输出

1.过程描述

1.1数据结构与算法

两个值得探究的问题

- 理论模型与实际性能的差异
- DSA优化的极限（下界

计算

- 计算既是我们的研究对象（规律、技巧），也是我们的目标（高效、低耗）
- 计算=信息处理，借助某种工具，遵照一定规则，以明确而机械的形式进行
- 所谓算法，即特定计算模型下，旨在解决特定问题的指令序列
 - 输入
 - 输出
 - 正确性
 - 确定性
 - 可行性
 - 有穷性
 - Hailstone sequence

算法：有穷性

$$\diamond \text{序列 } Hailstone(n) = \begin{cases} \{1\} & n \leq 1 \\ \{n\} \cup Hailstone(n/2) & n \text{ 偶} \\ \{n\} \cup Hailstone(3n+1) & n \text{ 奇} \end{cases}$$

$\diamond Hailstone(42) = \{ 42, 21, 64, 32, \dots, 1 \}$

$\diamond \text{int hailstone(int } n) \{ \text{//计算序列 } Hailstone(n) \text{ 的长度}$

$\text{int length} = 1; \text{//从1开始, 以下按定义逐步递推, 并累计步数, 直至 } n = 1$

$\text{while } (1 < n) \{ (n \% 2) ? n = 3 * n + 1 : n /= 2; \text{length}++; \}$

$\text{return length; \text{//返回 } |Hailstone(n)|}$

$\}$

$\diamond Hailstone(7) = \{ 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, \dots, 1 \}$

$Hailstone(27) = \{ 27, 82, 41, 124, 62, 31, 94, 47, 142, 71, 214, 107, \dots \}$

$\diamond \text{对于任意的 } n, \text{ 总有 } |Hailstone(n)| < \infty ?$

Data Structures & Algorithms (Fall 2013), Tsinghua University

4

■ 程序未必是算法

• 好算法：

- 正确（能处理简单、大规模的、一般的、退化的、任意合法的输入）
- 健壮（能辨别不合法的输入并做适当处理）
- 可读
- 效率（速度快，存储空间尽可能少）
 - $(\text{algorithm} + \text{data structure}) * \text{efficiency} = \text{computation}$

计算模型

- if you can not measure, you can not improve it
- 算法分析：
 - 正确性
 - 成本
- $T(n) = \max\{T(P) \mid |P|=n\}$ ，即在规模同为n的所有实例中，只关注最坏（成本最高）者
- 图灵机：

TM: Turing Machine

- 若当前状态为q且当前字符为c，则将当前字符改写为d；转向左侧/右侧的邻格；转入p状态
一旦转入特定的状态'h'，则停机

4

TM : Increase

- ### ❖ 规范 ~ 接口

\$

- 3

RAM: Random Access Machine

❖ 寄存器顺序编号，总数没有限制

//但愿如此

$R[0], R[1], R[2], R[3], \dots$

❖ 每一基本操作仅需常数时间

//循环及子程序本身非基本操作

$R[i] \leftarrow c$	$R[i] \leftarrow R[R[j]]$	$R[i] \leftarrow R[j] + R[k]$
$R[i] \leftarrow R[j]$	$R[R[i]] \leftarrow R[j]$	$R[i] \leftarrow R[j] - R[k]$
IF $R[i] = 0$ GOTO 1	IF $R[i] > 0$ GOTO 1	GOTO 1 STOP

❖ 与TM模型一样，RAM模型也是一般计算工具的简化与抽象

使我们可以独立于具体的平台，对算法的效率做出可信的比较与评判

❖ 在这些模型中

算法的运行时间 \propto 算法需要执行的基本操作次数

$T(n) =$ 算法为求解规模为 n 的问题，所需执行的基本操作次数

Data Structures & Algorithms (Fall 2013), Tsinghua University

6

RAM: Floor

❖ 功能：向下取整的除法， $0 \leq c, 0 < d$

$\lfloor c/d \rfloor = \max \{ x \mid d \cdot x \leq c \}$

$= \max \{ x \mid d \cdot x < 1 + c \}$

❖ 算法：反复地从 $R[0] = 1 + c$ 中减去 $R[1] = d$

统计在下溢之前，所做减法的次数 x

0	$R[3] \leftarrow 1$	//increment
1	$R[0] \leftarrow R[0] + R[3]$	//c++
2	$R[0] \leftarrow R[0] - R[1]$	//c -= d
3	$R[2] \leftarrow R[2] + R[3]$	//x++
4	IF $R[0] > 0$ GOTO 2	//if c > 0 goto 2
5	$R[0] \leftarrow R[2] - R[3]$	//else x-- and
6	STOP	//return $R[0] = x = \lfloor c/d \rfloor$

Step	IR	R[0]	R[1]	R[2]	R[3]
0	0	12	5	0	0
1	1	^	^	^	1
2	2	13	^	^	^
3	3	8	^	^	^
4	4	^	^	1	^
5	2	^	^	^	^
6	3	3	^	^	^
7	4	^	^	2	^
8	2	^	^	^	^
9	3	0	^	^	^
10	4	^	^	3	^
11	5	^	^	^	^
12	6	2	^	^	^

Data Structures & Algorithms (Fall 2013), Tsinghua University

7

2.结果输出

今天上午看了会书，下午摆烂，晚上才开始看数据结构的东西，也算是开始了。明天继续。