1.绪论

计算模型

RAM

There is an infinite set A that is not too big.

- J. von Neumann

邓俊辉

deng@tsinghua.edu.cn

Random Access Machine

❖ 寄存器顺序编号 , 总数没有限制: R[0] , R[1] ,

//但愿如此

//call-by-rank

- ❖ 通过编号,可以直接访问任意寄存器
- ❖ 每一基本操作仅需常数时间

IF
$$R[i] = \emptyset$$
 GOTO 1

STOP

IF
$$R[i] > \emptyset$$
 GOTO 1

//循环及子程序本身非基本操作

$$R[i] \leftarrow R[j] + R[k]$$

GOTO 1

Random Access Machine

- ❖与™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 ∞ 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

Floor Division

❖对任意 0 <= c 和 0 < d, 计算

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

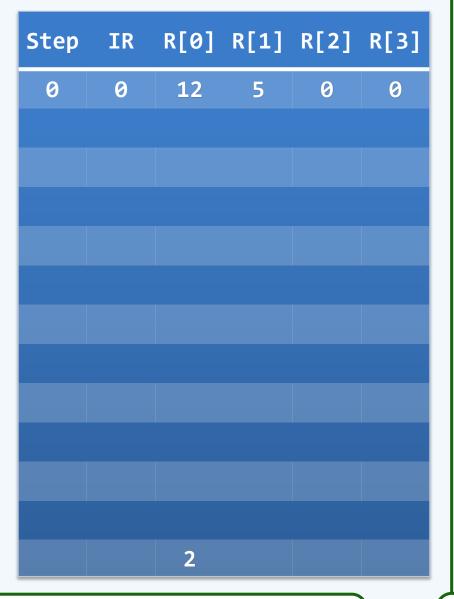
$$= \max \{ x \mid d \cdot x < 1 + c \}$$

$$\lfloor 6/3 \rfloor = 2 \qquad \lfloor 2016/36 \rfloor = 56$$

$$\lfloor 12/5 \rfloor = 2$$

❖思路: 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



Floor Division

❖ RAM算法

```
0 R[3] <- 1 //increment</pre>
    R[0] \leftarrow R[0] + R[3] //c++
    R[0] \leftarrow R[0] - R[1] //c -= d
    R[2] \leftarrow R[2] + R[3] //x++
    IF R[0] > 0 GOTO 2 //if c > 0 goto 2
    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	۸	٨	^	1
2	2	13	^	^	^
3	3	8	^	٨	^
4	4	^	^	1	^
5	2	۸	۸	^	^
6	3	3	^	^	^
7	4	۸	^	2	^
8	2	^	^	^	^
9	3	0	^	^	^
10	4	^	^	3	^
11	5	۸	^	^	^
12	6	2	٨	^	٨

课后

- ❖ 举例说明:随着问题实例规模增大,同一算法的求解时间可能波动甚至下降
- ❖ 在哪些方面,现代电子计算机仍未达到RAM模型的要求?
- ❖ 在TM、RAM等模型中衡量算法效率,为何通常只需考察运行时间?
- ❖ 图灵机算法Increase中,以下这条指令可否省略:

<0, #, 1, R, 1>

❖ 设计一个图灵机,实现对正整数的减一(Decrease)功能