CMPS 10 11-26-08

Insertion Sort

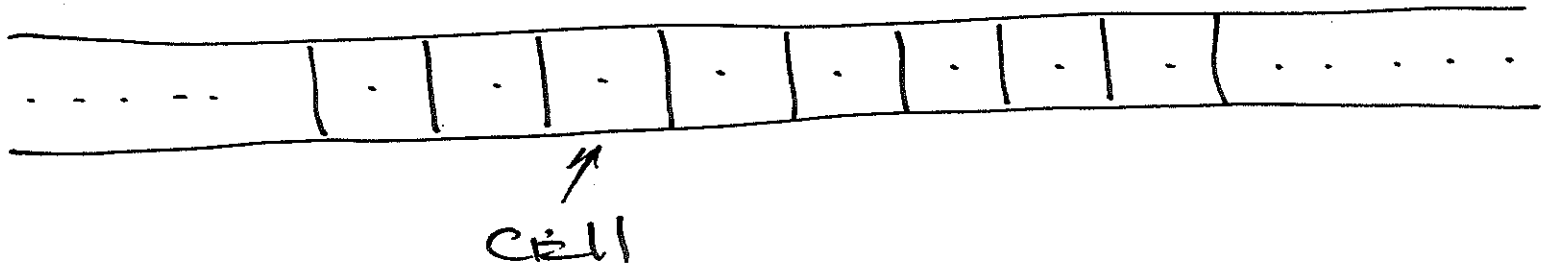
```
L ← 2
count ← 0
while L ≤ n
    j ← L
    while j ≥ 2 and (a_j < a_{j-1})
        count ← count + 1   ← TEST WAS TRUE
        a_j ↔ a_{j-1}
        j ← j - 1
    if j ≥ 2
        count ← count + 1   ← TEST WAS FALSE
    L ← L + 1
stop
```

Counting Comparisons in Lab 5

TURING MACHINES

- TAPE

REPRESENTS: { INPUT, OUTPUT, MEMORY }



EACH CELL CONTAINS 1 SYMBOL, FROM AN FINITE SET CALLED THE ALPHABET

$$\{\underbrace{b, 0, 1}_{\text{USUALLY INCLUDED}}, \ldots\}$$

ALL BUT FINITELY MANY SYMBOLS ON TAPE ARE b (blank.)

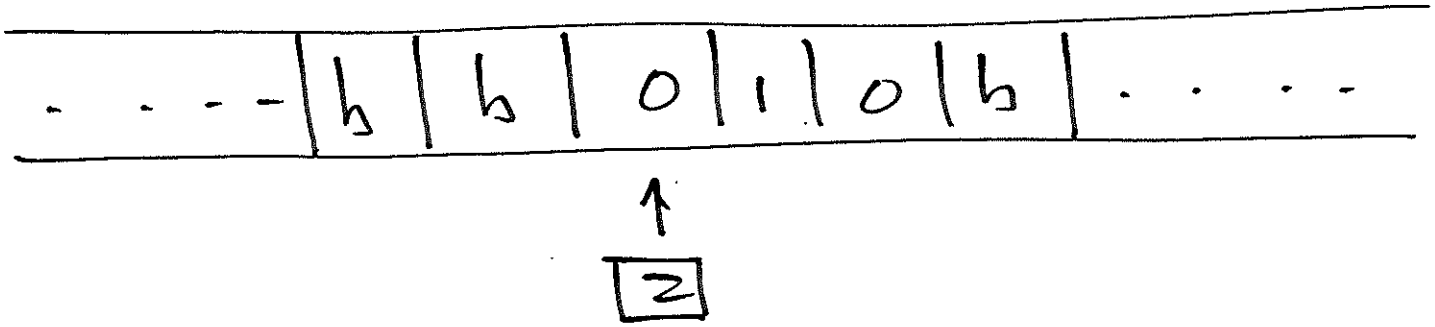
- A DEVICE FOR READING & WRITING SYMBOLS TO THE TAPE CALLED THE HEAD (READ HEAD, RW-HEAD).

- A FINITE SET OF INTERNAL STATES REPRESENTED BY POSITIVE INTEGERS

$$\{1, 2, \ldots, n\}$$

THE STATE CAN CHANGE, DEPENDING ON WHAT CHARACTER IS CURRENTLY BEING READ.

Ex.



THIS IS A PICTURE OF A TURING MACHINE WHICH IS IN STATE 2, READING CHARACTER '0'.

- A TURING MACHINE PERFORMS EXACTLY 1 PRIMITIVE OPERATION, CONSISTING OF 3 ACTIONS

1.) WRITE A SYMBOL (OVERWRITING WHAT IS ALREADY THERE.)

2.) GO TO A NEW STATE (COULD BE SAME STATE.)

3.) MOVE THE HEAD LEFT OR RIGHT BY 1 CELL.

WHICH STATE YOU GO TO & WHICH SYMBOL YOU WRITE DEPENDS ON CURRENT STATE & SYMBOL.

THUS A TURING MACHINE INSTRUCTION IS A CONDITIONAL statement.

If (in state i) and (reading symbol j)

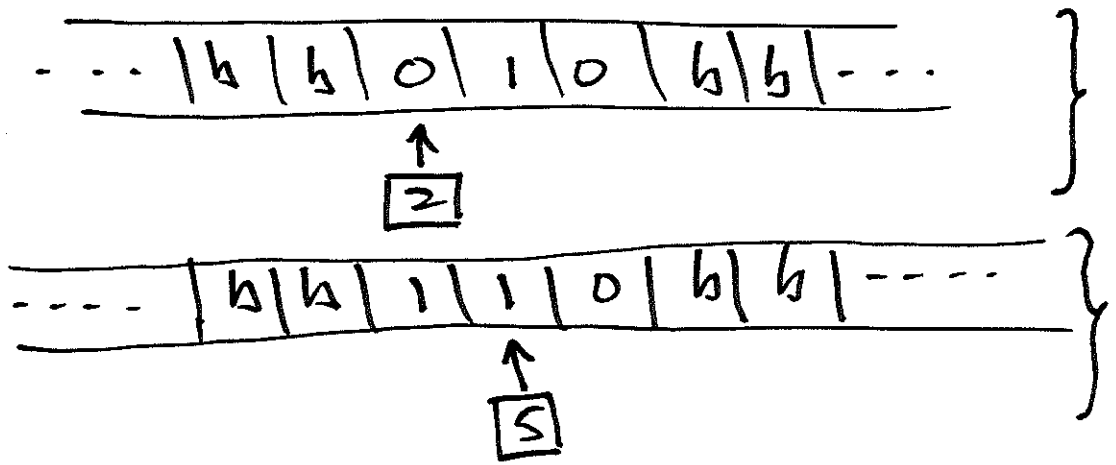
write symbol k
go into state s
move in direction D

ABSTRACTLY, WE REPRESENT THIS AS A 5-TUPLE

(i, j, k, s, D)

EX. $(2, 0, 1, 5, R)$

CONFIGURATIONS



A <u>CONFIGURATION</u> is A COMBINATION OF state, tape, and read-head position on tape.

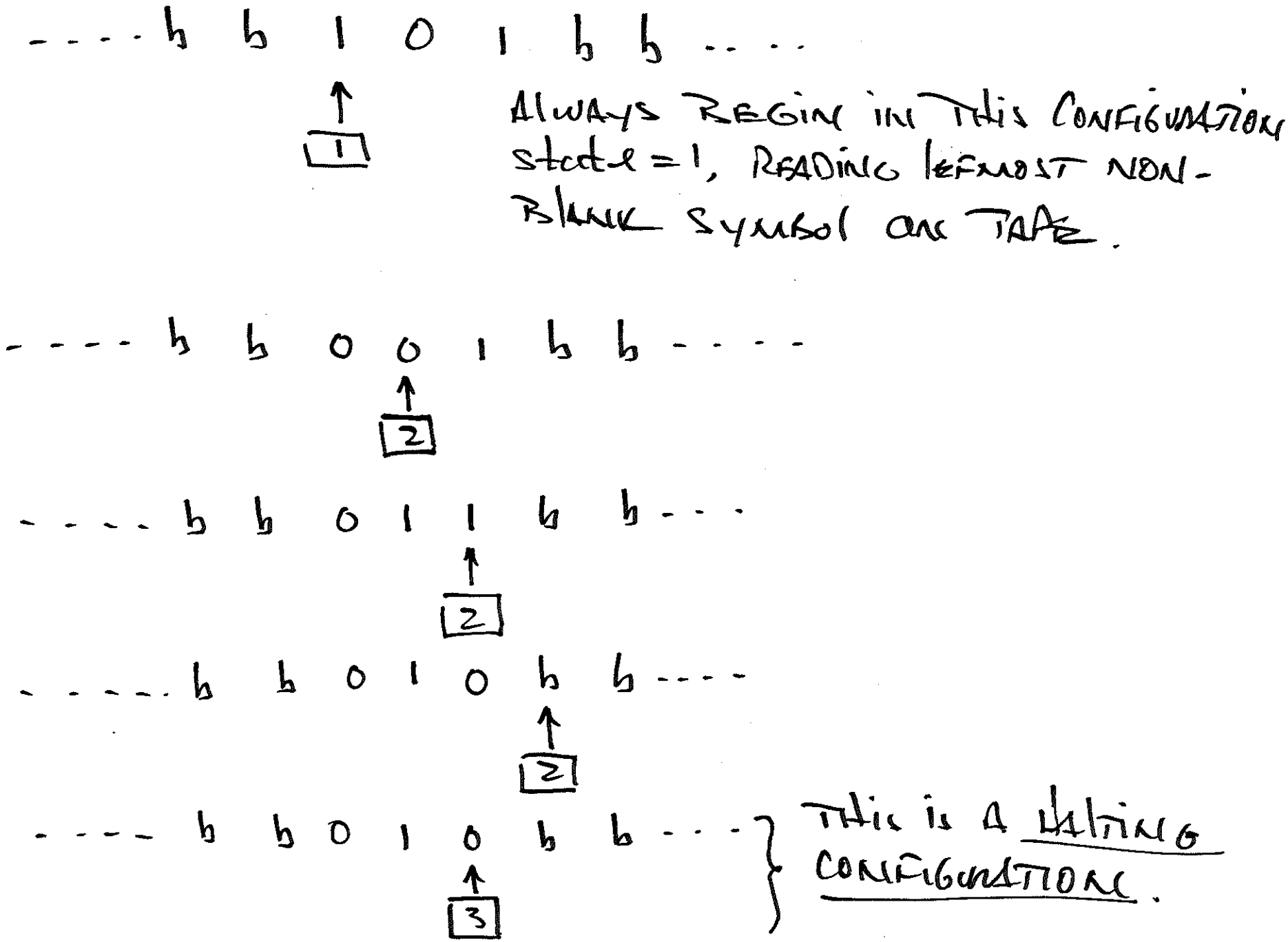
⇒ (2, 0, 1, 5, R) TRANSFORMS CONFIG. I. INTO CONFIG. II.

A PARTICULAR TURING MACHINE is JUST A FINITE COLLECTION OF SUCH 5-TUPLES.

<u>EX</u> (1, 0, 1, 2, R)
(1, 1, 0, 2, R)
(2, 0, 1, 2, R)
(2, 1, 0, 2, R)
(2, b, b, 3, L)

ALPHABET:
{b, 0, 1}

TAPE:


. . . . b b 1 0 1 b b . . . .
1
ALWAYS BEGIN IN THIS CONFIGURATION
state = 1, READING LEFTMOST NON-
BLANK SYMBOL ON TAPE.
. . . . b b 0 0 1 b b . . . .
2
. . . . b b 0 1 1 b b . . .
2
. . . . . b b 0 1 0 b b . . . .
2
. . . . b b 0 1 0 b b . . .
3
THIS IS A HALTING
CONFIGURATION.

NOTE: THERE IS NO INSTRUCTION OF THE FORM

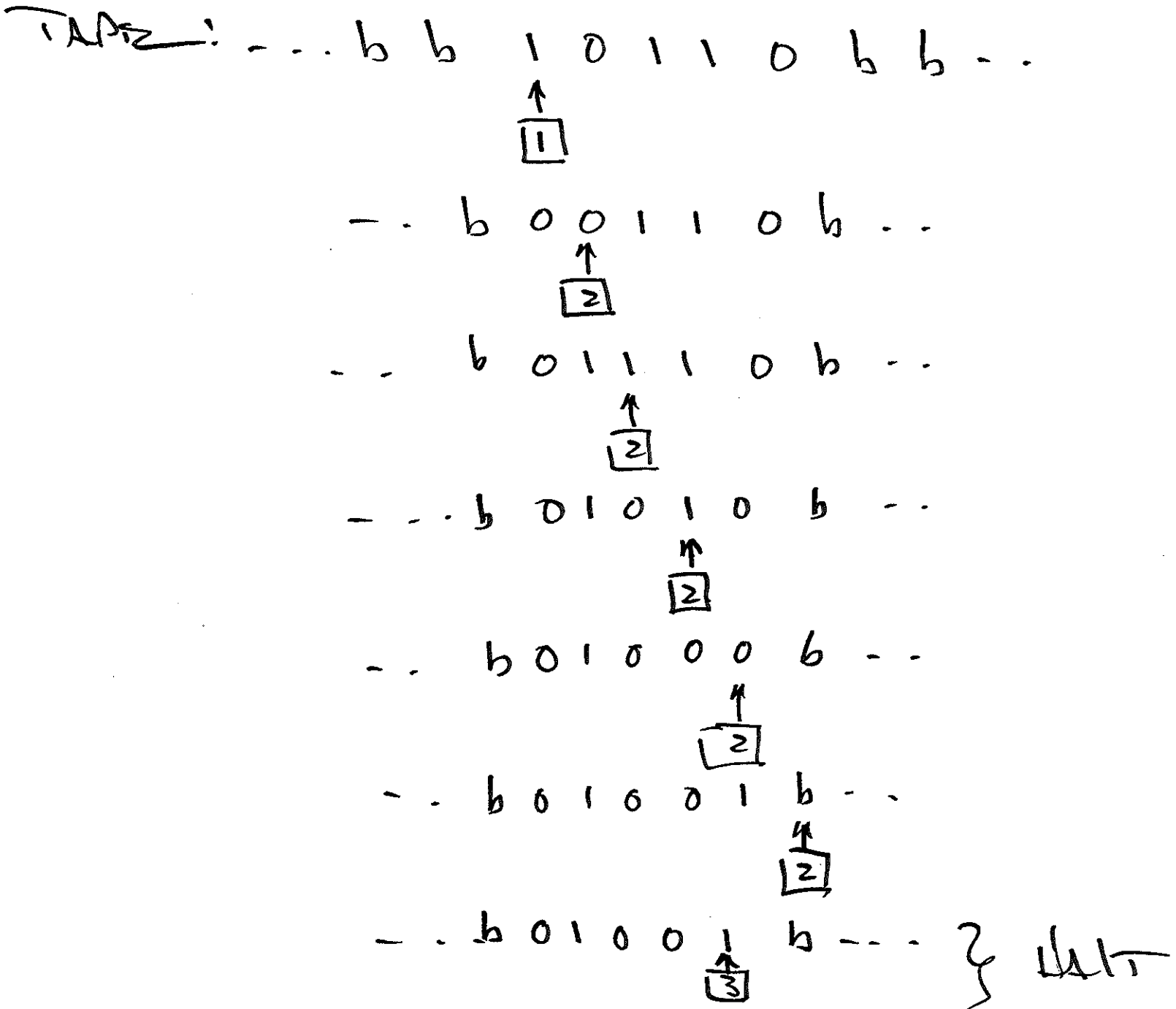
$$(3, 0, \cdot, \cdot, \cdot)$$

SO WE HALT.

NOTE: OUR TURING MACHINE CANNOT HAVE TWO DISTINCT INSTRUCTIONS OF THE FORM

$$\begin{array}{ccccccc} (i, & j, & c_1, & s_1, & D_1) \\ \| & \| & + & + & + \\ (i, & j, & c_2, & s_2, & D_2) \end{array}$$

Ex. Skip numbers



Evidently this machine is a bit inverter.

Ex. Other machines

--- b0010b1101b001b---