CMSC 10 12-2-10

Chap 12: Models of Computation

Turing Machines

languages we've studied:

- Pseudo-code
- Circuits
- C++

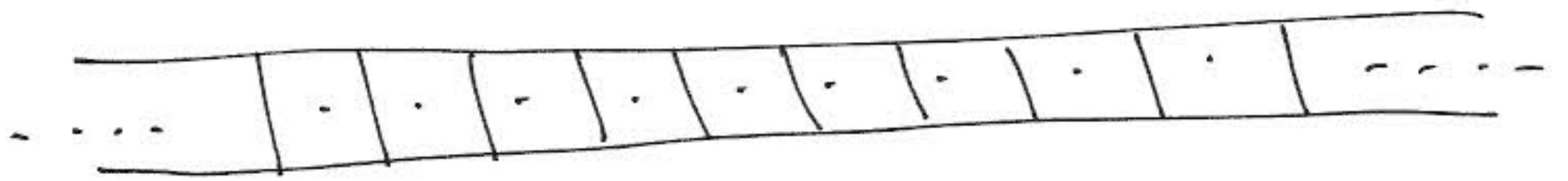
Recall Defn of Algorithm:

Defn: A collection of operations which are

- well ordered
- unambiguous (Primitive)
- EFFECTIVELY Computable
- Produce a result
- Halt in finite time

A Turing Machine includes

- A Tape which extends infinitely in both directions.



Divided into cells, each holding

1 symbol.

Symbols belong to a finite set A called Alphabet.

special symbol: Blank "b"

usually

$A = \{ b, 0, 1, \ldots x, y, \ldots \}$

The Tape

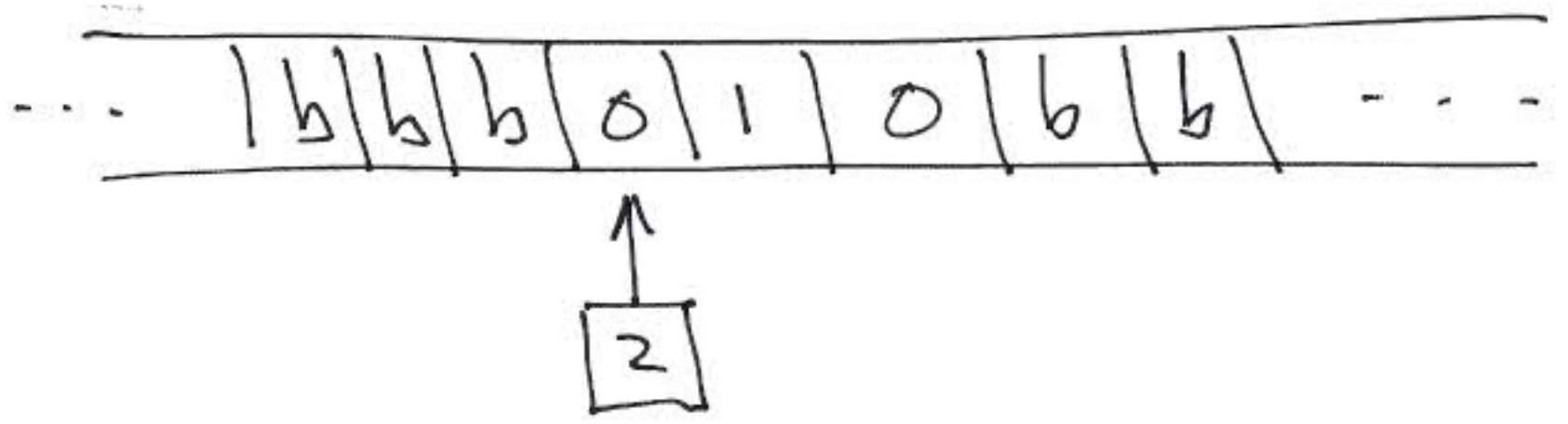
- stores input data
- serves as working memory
- stores output data

At any time, the tape stores only finitely many non-blanks.

- A DEVICE for reading from and writing to the tape : HeaD

- A finite set of states $\{1, 2, \ldots, n\}$

Ex.



Rule: Tape Head Always starts at leftmost non-Blank Symbol.

- Performs one Primitive Operation
  1.) write a symbol in a cell (overwriting symbol already there)
  2.) go into a new state (could be same state.)
  3.) move head either 1 cell Left or 1 cell Right.

Form of a Turing machine instruction

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

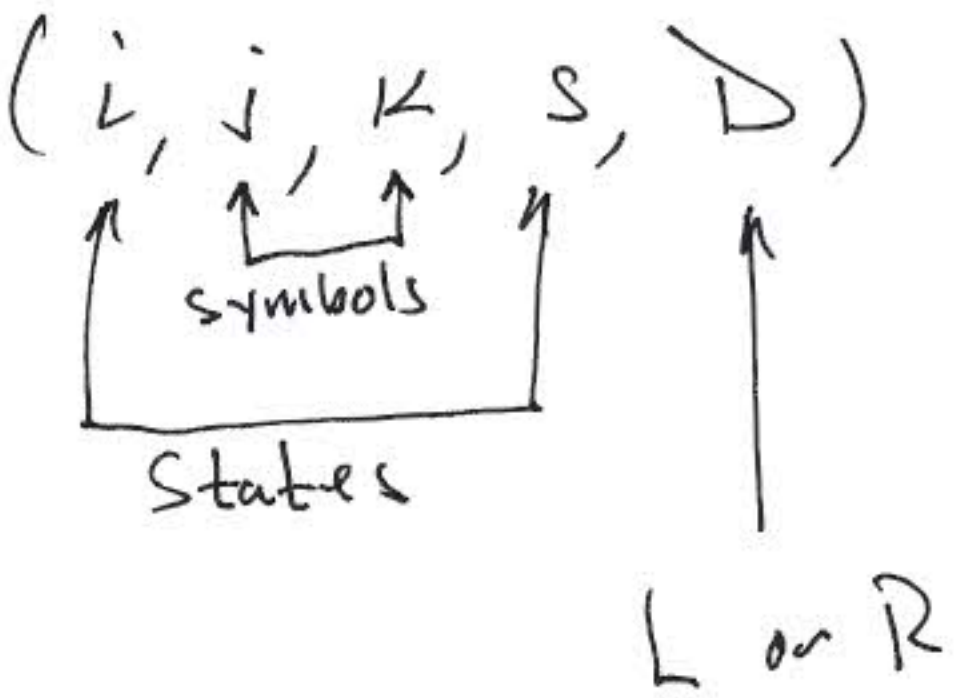
write symbol K

go into state S

move in direction D

Such an instruction is specified

By a 5-tuple



Ex. states = {1, 2, 3, 4, 5, 6}

$A = \{b, 0, 1\}$

Instruction: (2, 0, 1, 5, R)

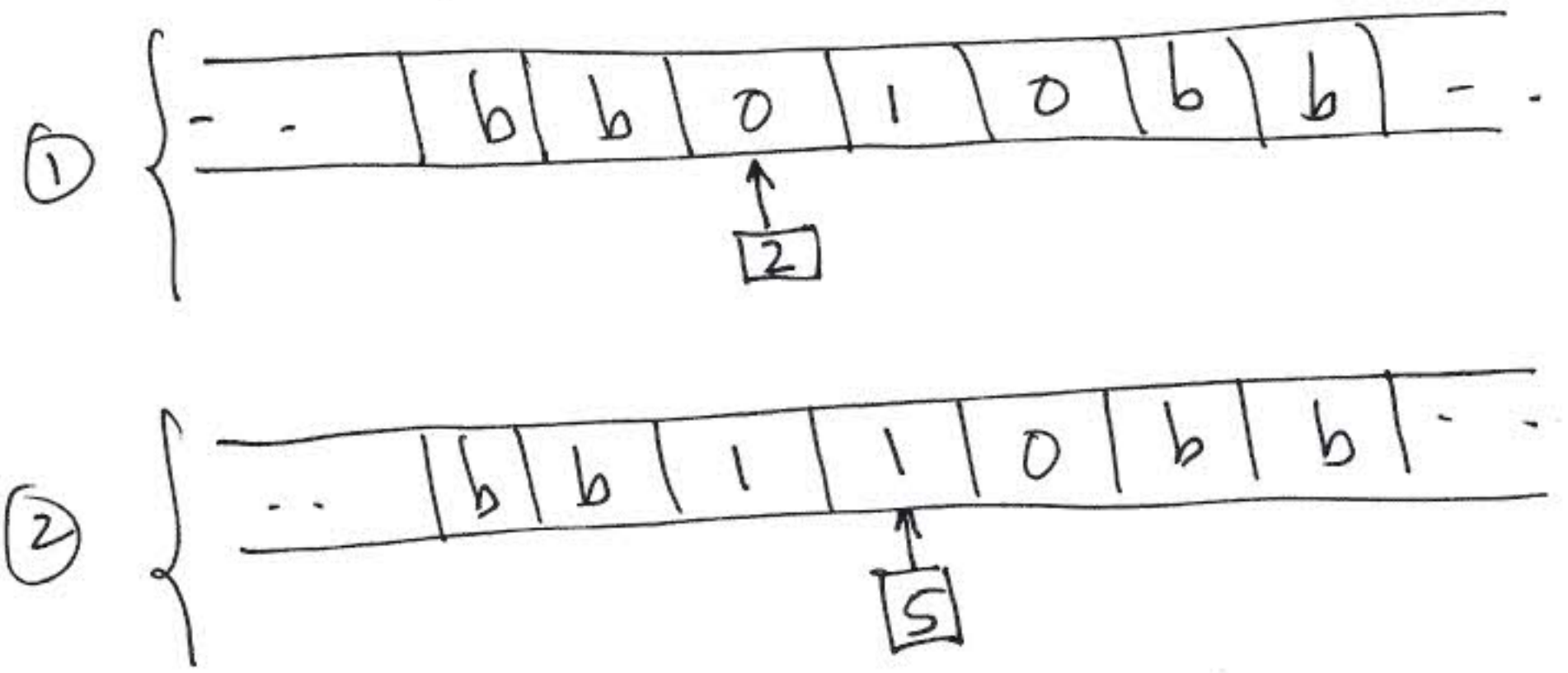
Says

if in state 2 reading 0

write 1

go to state 5

move R



A combination of: Head Location and state, is called a configuration.

A Turing Machine (or a Turing Machine Program) is a finite set of such instructions

Ex. (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)

Also called a Transition Function

states = {1, 2, 3}, a = {b, 0, 1}

Rule: start in state 1,
reading leftmost non-blank

Ex. ... b b 1 0 1 b b ....

... b b 0 0 1 b b ....

.. b b 0 1 1 b b ...

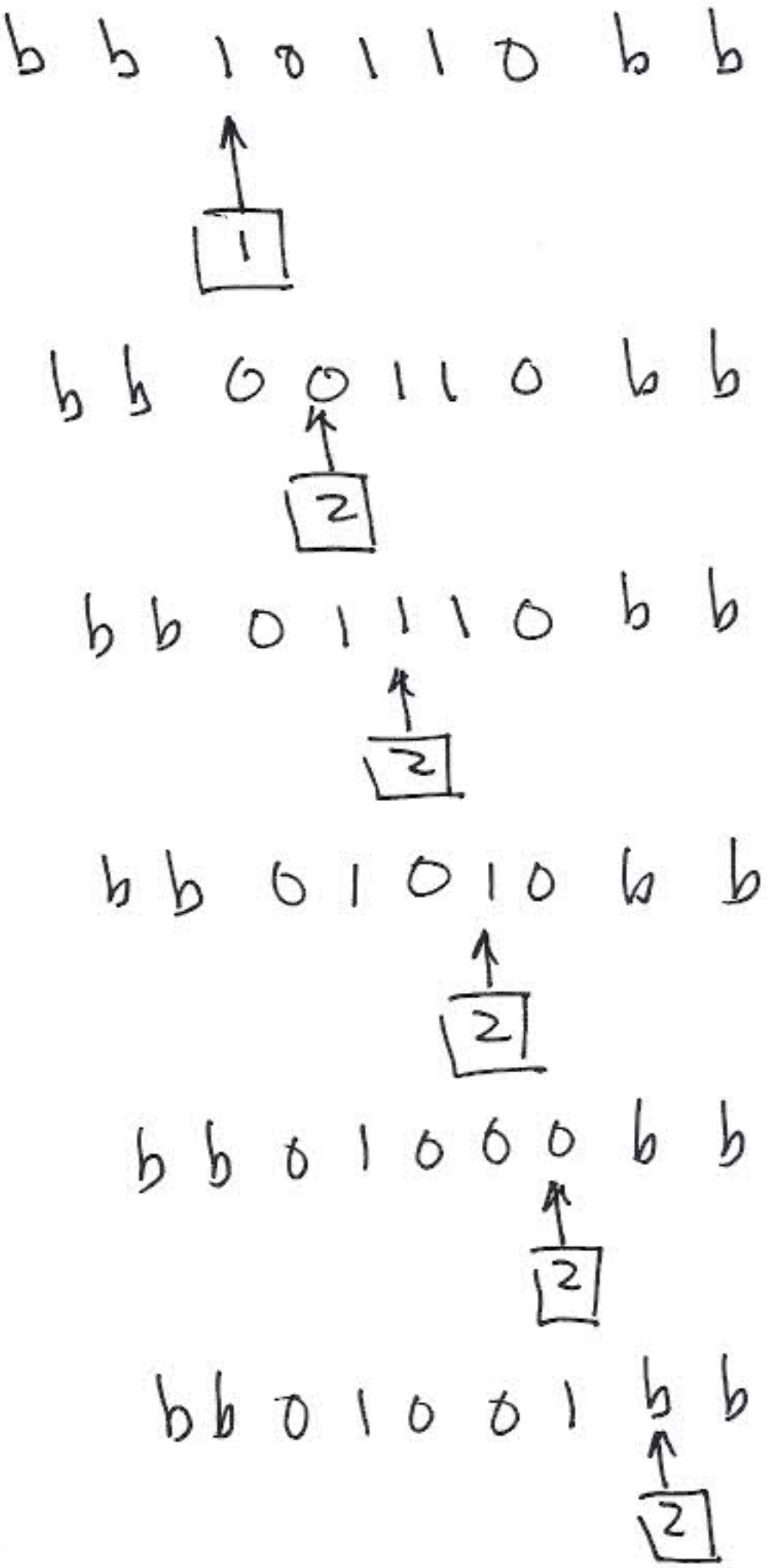
... b b 0 1 0 b b ..

... b b 0 1 0 b b ...

Halting Configuration

Have no instruction: (3, 0, ·, ·, ·)

Ex. Same Turing Machine
Different Tape



b b 0 1 0 0 1 b b

↑

[3]

Halting Configuration.

Exercise:

Same T.M., new Table

- - b b 0 0 1 0 1 b b - -

↑

[1]

Ex. another T.M.

(1, 0, 1, 1, R)

(1, 1, 0, 1, R)

Try this on

b b 0 0 1 0 1 b b

Correspondance:

Algorithm $\longleftrightarrow$ Turing Machine

Instance of Problem $\longleftrightarrow$ Tape

Ex ODD Parity Machine

(1 1 1 2 R)

(1 0 0 1 R)

(2 1 1 1 R)

(2 0 0 2 R)

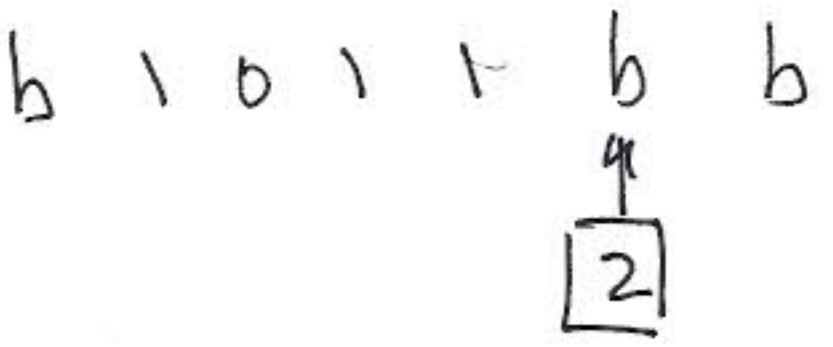
(1 b 1 3 R)

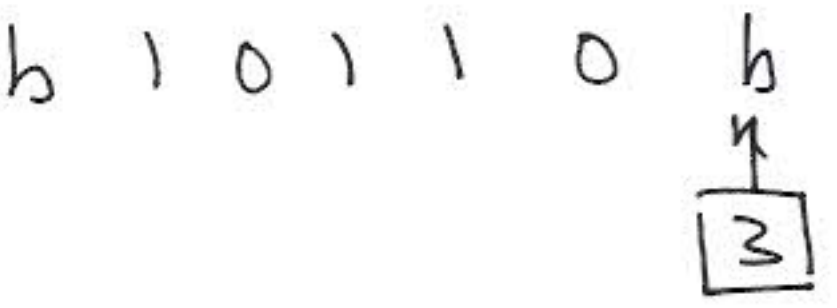
(2 b 0 3 R)

Try on: b 1 0 1 1 b

b 1 0 1 1 b

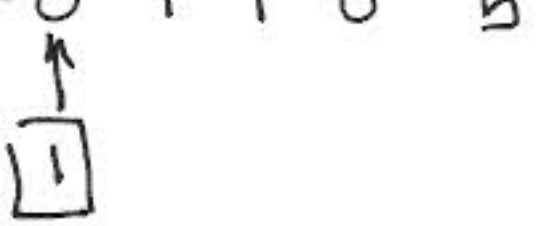
b 1 0 1 1 b

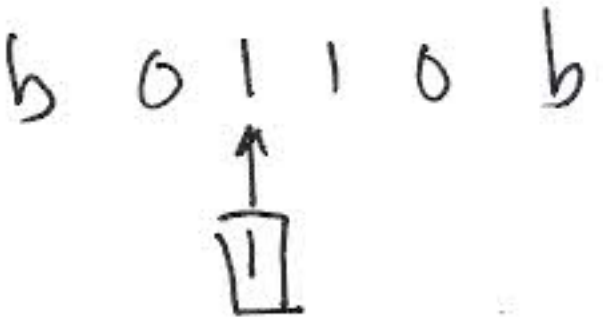


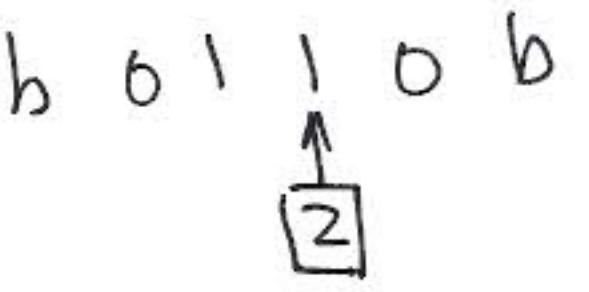


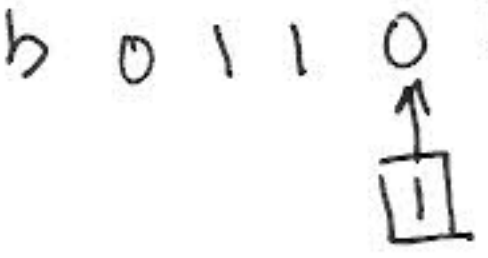
Halt

Try on: b 0 1 1 0 b









b b 0 1 1 0 b b

↑
1

b b 0 1 1 0 1 b

↑
3

Halt

Ex. Binary Increment

1	1	1	1	R
1	0	0	1	R
1	b	b	2	L
2	0	1	3	L
2	1	0	2	L
2	b	1	3	R

check: initial b 1 0 1 b

final b 1 1 0 b