

$$(\star X) + 2$$

↑

$$X = [3][4][5][10]$$

$$\star X = [4][5][10]$$

$$(\star X) + 2$$

↓

$$[\star][2][0][0]$$

↑

$$[1][2][0][0]$$

↓

COMPRESS STRING

aaabbbccccccaa




The diagram illustrates the compression of the string "aaabbbccccccaa". The first 'a' is enclosed in a black box. Three orange arrows originate from this box: one points to the second 'a', another to the third 'a', and a longer one to the first 'b'. Similarly, an orange arrow points from the first 'c' to the last 'c'. These arrows represent the run lengths of the characters.

// PRB



S = STRINGA DI
CARATTERI (\emptyset)

RESULT =



STRINGA CHE COMPRESSE

C I A O



C 1 1 1 A 1 0 1

$\forall C, C = \text{CARATTERI}$

$0 \leq i \leq (M)$

LUNG.

significati DAL # DI STRINGA
OCCORRENZE

$\emptyset \rightarrow \emptyset \leq \emptyset \leq (M)$

POST

RESULT



C I A O \rightarrow C 1 1 1 A 1 0 1

$\forall C, C[i-1] \neq C[i]$



CARATT. VI CM
SONO OMNISI

$\forall C, N. \text{ DI OCCORRENZE}$

$\forall i, j \in S$

$C[i] = C[j] \Rightarrow$

PERREBBONO

SSSIS UGLI

$C \dots C$

\uparrow

CONT FINE

NON APPARE UN

CARATT. D MORSO

\downarrow

COMPRESSIO

COMPRESS - STRING

↓
1 RETURN

✓ IF (S == NULL) {
 RESULT = NULL;
 RETURN;
}

* S == "0"
 ↑
 VVVV

INT COUNT = 0;

↓
[A A B B <<< A]

FOR (i = 0; i ≤ S.LENGTH;
 i++)

CHAR * CURRENT = S[1]

CURRENT (* S)
[A A B B < < < A]

WHILE (CURRENT == S[1]) {

COUNT++

S++ ;

[A A B B < < < A]

2
}

RESULT[1] = CURRENT

RESULT++ A 2

RESULT[1] = COUNT

[A 2] B B < < < A

AAB
 ↓

$A \neq B$

A2

↓ INT (RECURS)

COMPRESS_STRING

[AABB<<A]

* CURR = *S

IF (S == CURR) {
 ✓ COUNT =
 COMPRESS(S+1, RESULT)
 }

* RESULT = *CURRENT

RESULT++

* RESULT = COUNT

[1] [2] / [3] [4] [5]

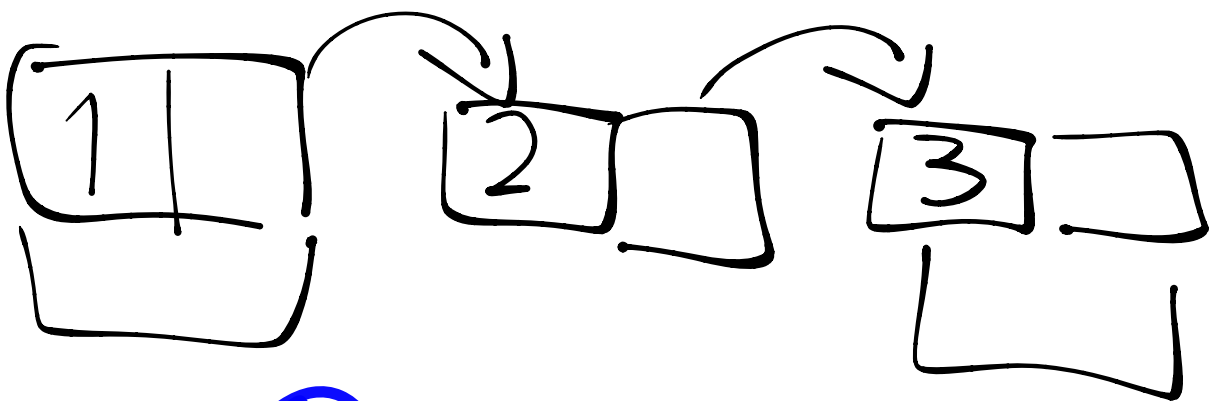
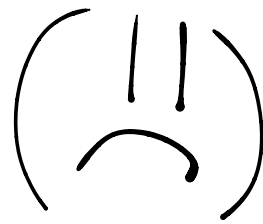
↑ ↑
MS MD

[5] 2 3 4 [1]

SCALBIA

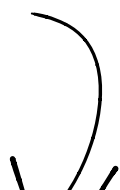
ОБЪЕМ А

CORPUS



① < 3

↘ FIRST-ODD



LISTA * FIRST-000

(LIST A * SRCPTR){

IF (SRCPTR == NULL)

RETURN NULL

LISTA * MIN =

* SRCPTR;

① 2 ③ 4
↑

WHILE (SRCPTR) {

IF (SRCPTR->NR0

% 2 == 1 0000

SRCPTR->NR0

< MIN->NR0

MIN = SRCPTR

}

SRC = SRC → NEXT

}

RETURN MIN

/* IF (SRC) {

SRC = FIRST-OD
(SRC → NEXT)

IF (SRC < MIN)

MIN = SRC

RETURN MIN



STAGE 5 - IT

INT POSSA

(INT CAPRO[5][3],
POS_X, POS_Y)

IF(POS_Y == DUT_Y)

RETURN 1; / POS_X ≤
DUT_X

RETURN POSSA

(CAPRO[DUT_X][DUT_Y],
POS_X+1, POS_Y)

+

RETURN POSSA

(CAPRO[DUT_X][DUT_Y],
POS_X, POS_Y+1)

$$pos = x, pos_{y+1})'$$

