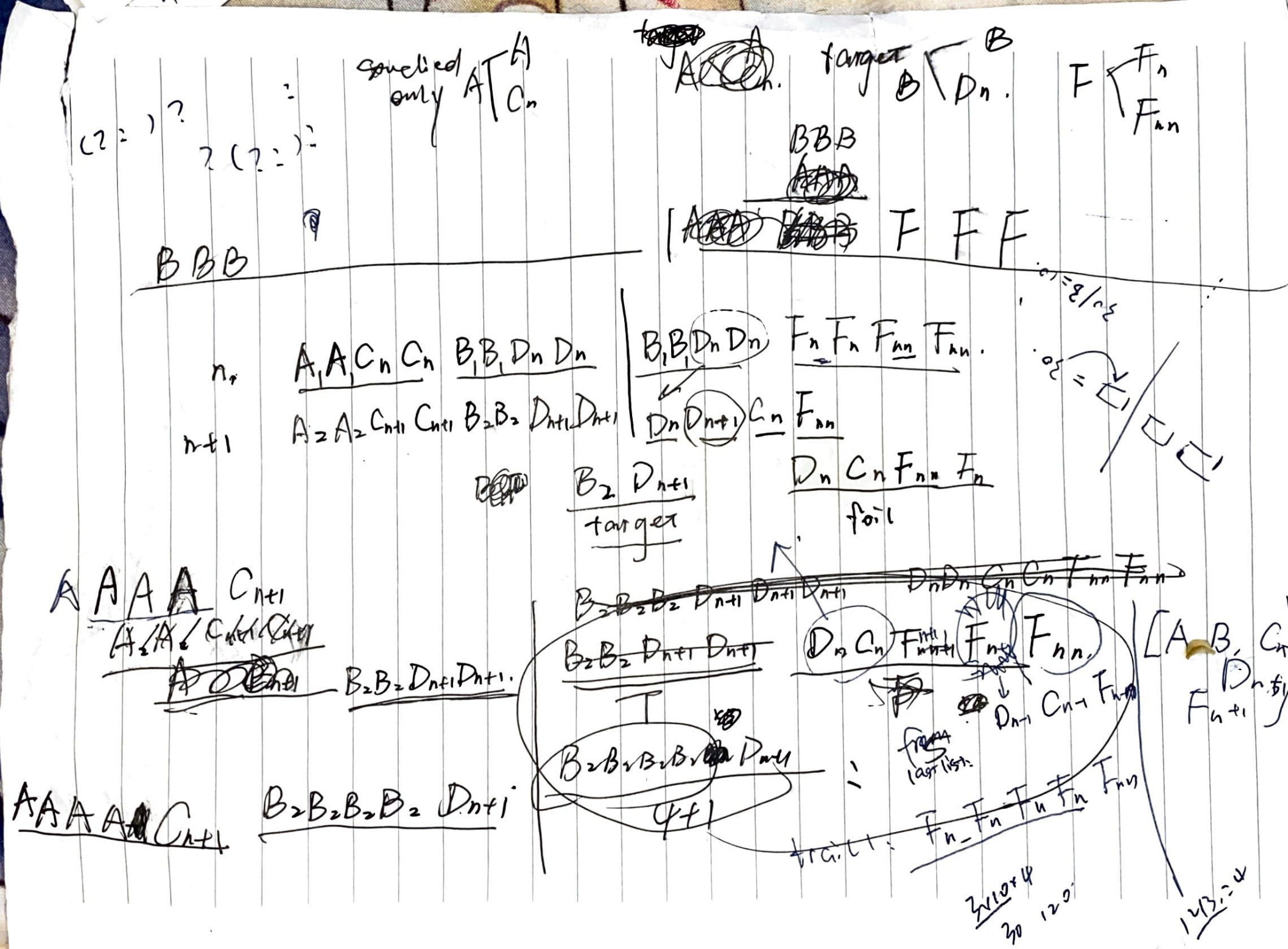


固定 — A = 120 — [12, 12, …12]
B = 120
⋮
→ initial Structure: Fn 49 [12, 3, 3 … 3]

C_{n+1} ABC$_{n+1}$D$_{n+1}$ 不继承
list 1: B D$_{n+1}$ (F$_{n+1}$, F$_{n+1}$ F$_{n+1}$, F$_{n+1}$) F$_{nn+1}$
(C$_{n+1}$ ↓ 1
"f" "r": 全部 assign F test position (15 '1)
"b":
(A, B) F$_{n+1}$ study 时 assign F test position.

ABC$_n$D$_{n+1}$ 从上继承 不继承 会往下继承
list 10: B . D$_{n+1}$ (D$_n$ C$_n$ F$_{nn}$ F$_{n+1}$ F$_{nn+1}$)
"f" "r": (A, B, C$_{n+1}$ D$_{n+1}$) + B, new pos.
"b": 全部 new pos

根据继承多少个 list 每类数量

		A	B	C_{n+1}	D_{n+1}	F_{n+1}	F_{nn+1}
Initial Test list 1:		A: 4	B: 4	C_{n+1}: 1	D_{n+1}: 1	F_{n+1}: 4	F_{nn+1}: 1
Truth	"f" Few: 4	4	4	1	1	4	1
Final Test Assign	"b"	4	4	X 下放	X 下放	4	X

list 3 (2):

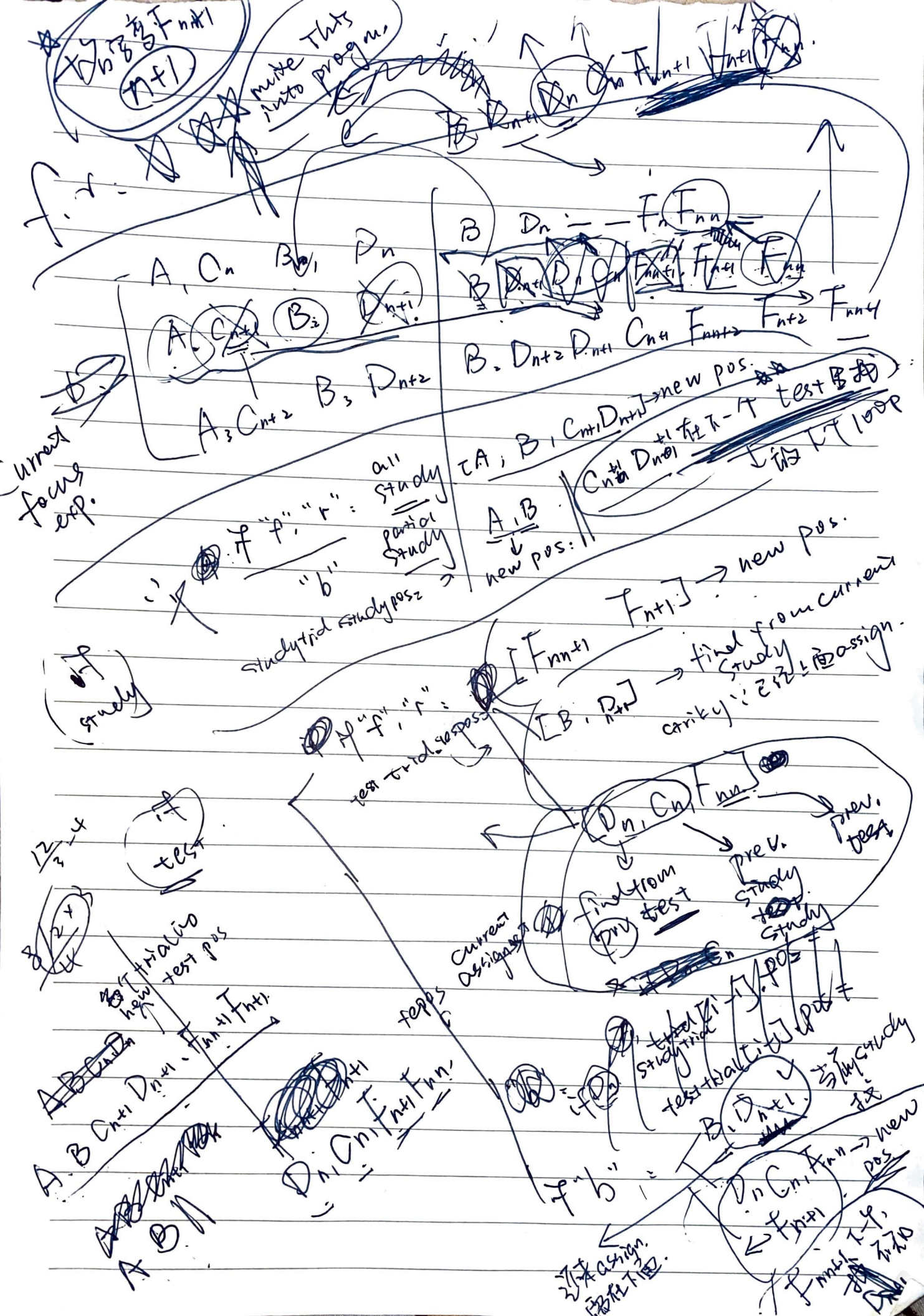
D_n	C_n	F_{nn}	A	B	C_{n+1}	D_{n+1}	F_{n+1}	F_{nn+1}
D_n	C_n	F_{nn} = 1	A: 4	B: 4	C_{n+1}: 1	D_{n+1}: 1	F_{n+1}: 1	F_{nn+1}: 1
X 上放	X 上放	X 上放	4	4	1	1	1	1
1	1	1	4	4	1	1	1	1

"b" list

list 2 list 10:

D_n	C_n	F_{nn}	A	B	C_{n+1}	D_{n+1}	F_{n+1}	F_{nn+1}
D_n 1	C_n 1	F_{nn} 1	A 4	B: 4	C_{n+1}: 1	D_{n+1}: 1	F_{n+1}: 1	F_{nn+1}: 1
1 上放	1 上放	上放	4	4	1	1	1	1
1	1	1	4	4	下放 (+1)	下放 (+1)	1	下放 (+1)

"f" "r"



如果要把所有的写在一个 for loop 中如何实现：

A C_n B D_n | B D_n F_n F_n F_n F_n F_{nn}

A C_n B_0 D_n | B D_n — — F_n F_{an} —

A C_{n+1} B D_{n+1} | B D_{n+1} D_n C_n F_{nn+1} F_{n+1} F_{nn}

A C_{n+2} B D_{n+2} | B D_{n+2} D_{n+1} C_{n+1} F_{nn+2} F_{n+2} F_{nn+1}

appear2_testpos:

if study: C_{n+1} D_{n+1}. next find.

if test: D_{n+1}, E_{n+1}, next find.

不知道下一个 assign 下的 ... 要实时回去 assign.

如果: current trial = n+1

if current trial

X_{n+1}

x^2+1 ... $2x^2+2=0$. 去实时 ... 倒

比如 ex: i = n+2

obj. C_{n+1} - curr test = 4 test:

obj. C_{n+1} - curr study = 0.

i = n+2

obj. F_{n+1}

if [D_{n+1}, C_{n+1}] include [ ]

Last obj. appear2 testpos

= curr obj. testpos

Last obj.

480

i.e.,

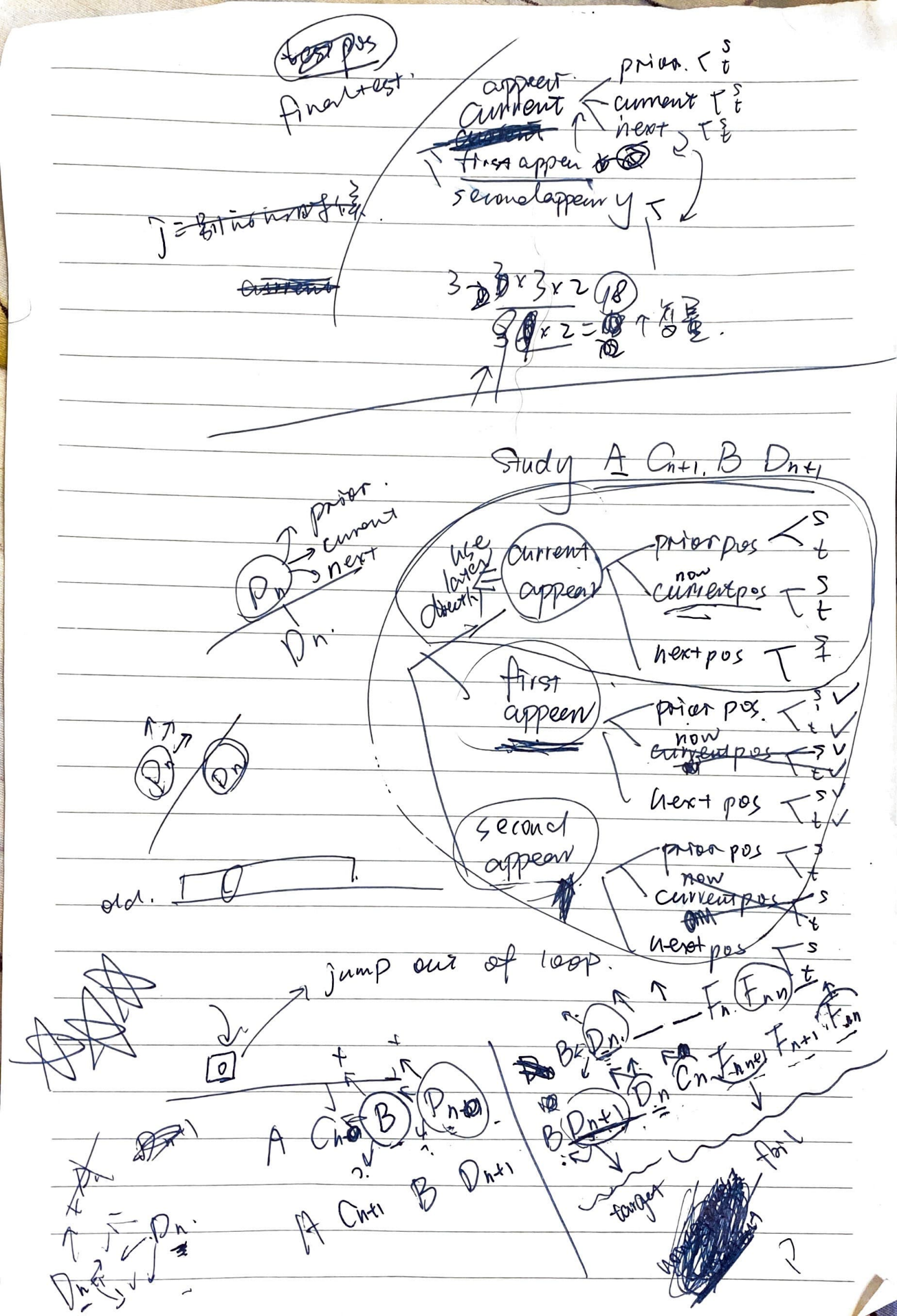
i = n+1.

[D_n, C_n, F_n]

$A: 4 \times 3 \times 10$

$= 120$

$12 \times 1 + 4$


Study A C_{n+1}, B D_{n+1}
prior pos
now current pos
next pos
current appear
first appear
second appear
jump out of loop.
target
fail

*check if final position assignment 靠下 一半 15×2 30个 ×2=60个

偶 0, 2, 4, 6, 8, 10

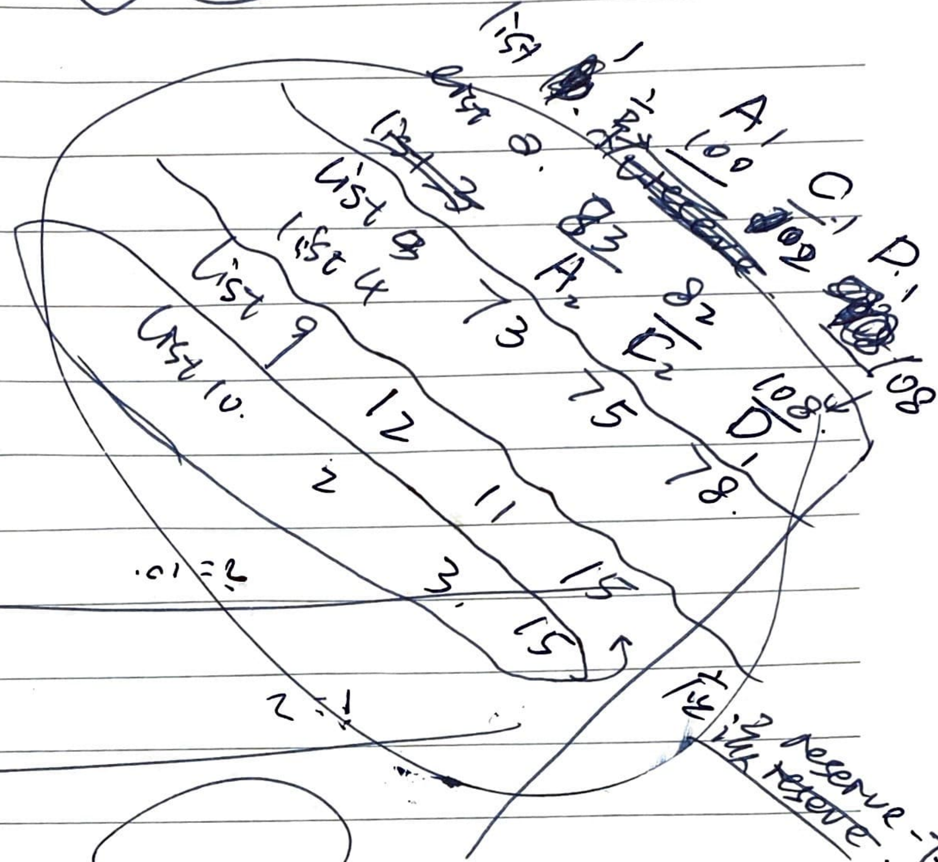
1 3 5 7 9 奇

偏 8 ✗

list 9

list 10 i= 9 ✓

i=2



col=2

i=1

col=2

for

[2, 1, 5]

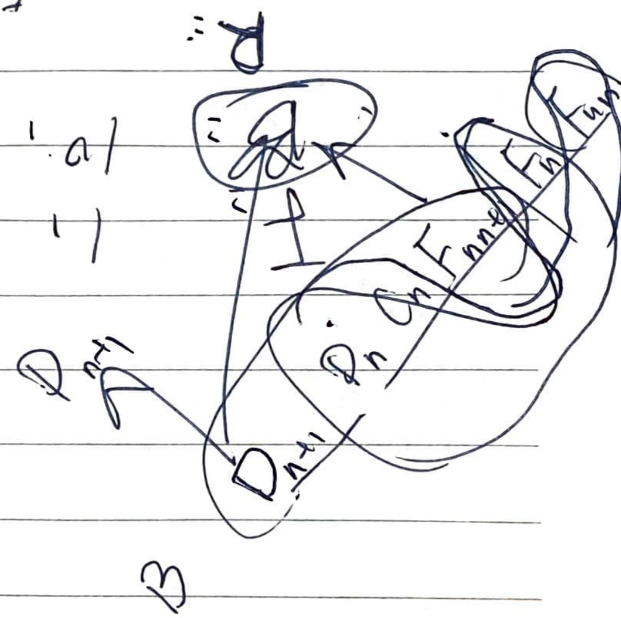
[1,2, ... ,10]

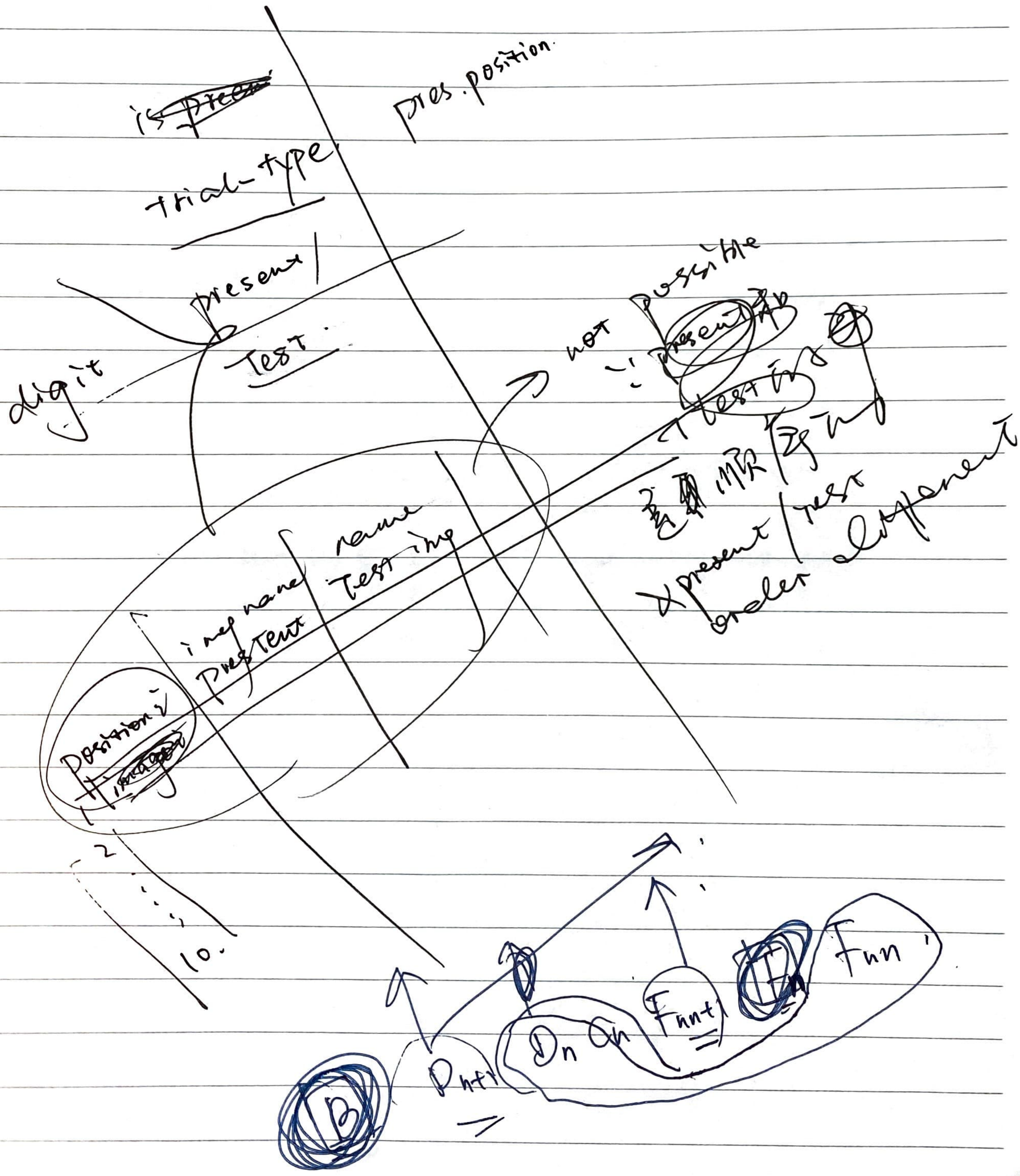
[3, 1, 5]

[1, 0]

j=1

i=




pres. position
trial_type
present
Test
digit
position
imgname present
name Testing
not possible
present
test
present | test
order different
1
2
10
Fun
B

→ Final feedback:

only recall if it has been studied or tested previously but not more info

→ initial: → Correct ↔ Incorrect

700 510 = 306.

(204)

Meaning Content:

~~rehearsal keeps accumulating~~

~~buffer size 3.~~

~~negative recency~~ (recency not buffering)

arithmetic cancel stop rehersive

A: $120 \times \frac{2}{3}$ [1 1 1 2 2 2 ...]

splice (0, 120).

B:

D: $30 \times \frac{2}{3}$ [...]

splice (0, 30)

$12/3 = 4$

$120 + 120 + 30 \times 4 + ?$

$= 240$

$+ 120$

$+ 89$ (7 respos = 6)

$369 \times \frac{2}{3}$

123

$= 246$ final for

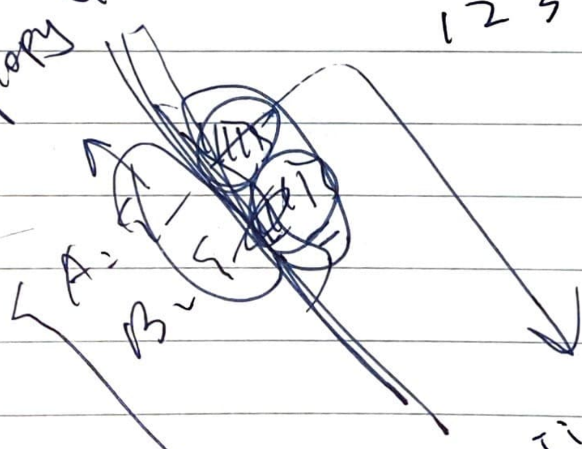
"4" = list 1 15 个

"0" = list 10 15 个

item_id

[283] list 1 ... list n

deepcopy



0-30

31 + 24 = 55

[n]

: [i, ...

list 1 15 个 A 30 B 30 12 个

list 0 ... 10

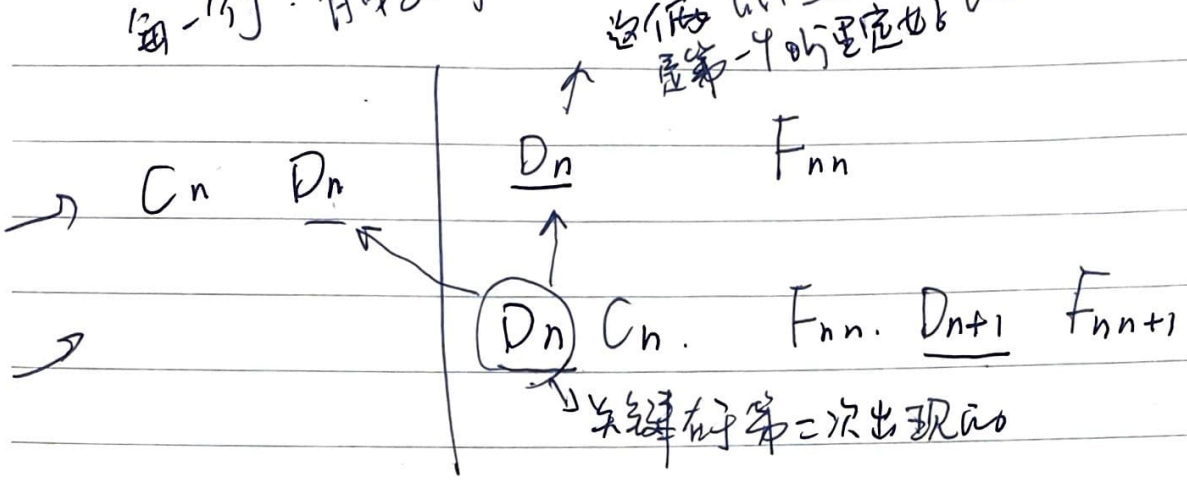
assign initial 15 个 ... 3 + 2

initial ... new randnames.map(..., ...)

A: ... map

[illegible]

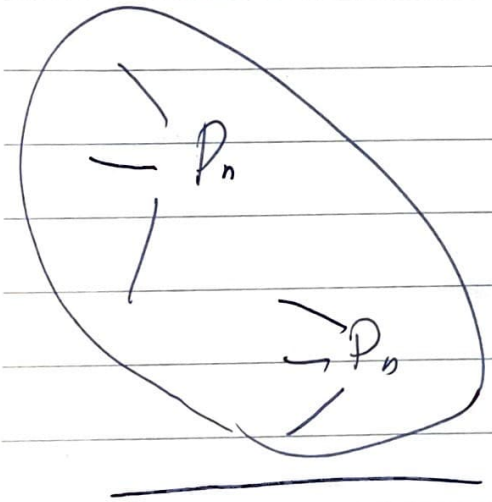
每一行. 有重复的.



Dn: list ⇒ [ABCD] 不重复

[ ]

如果 Dn 前 Dn →
1. repeat item
2. 但 no prior pos
current no. test/study
3. Study pos, test pos
4. next study pos, test pos



后 Dn →
1. repeat item
2. has prior pos
test or study.
3.

有错误性

☆ 无错误性

S: A B Cn Pn

被错误的

T: B D Fn Fnn Dn-1 Cn-1 Fnn-1

X ← Ⓐ ← Ⓐ A

studypos_initial

B D

A C | B D DCFFF

study | test

这些只写到data里不写的里

prior secondappear studypos testpos_current

1 2 2 8

study list 1 A

2 8

test A

exist 1 ——

exist 2 A
Test

A

A

?(? =) =

:(? :) undefined.

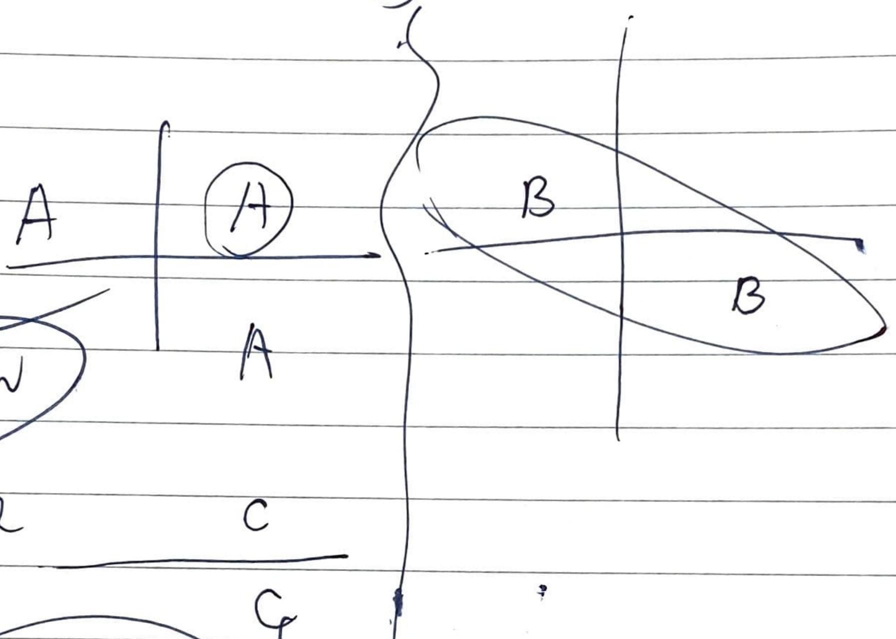
?

冲突点:

① A ~~两~~ → two test position

②

first appear



second appear ← valid only when repeated.

second

~~study_current~~

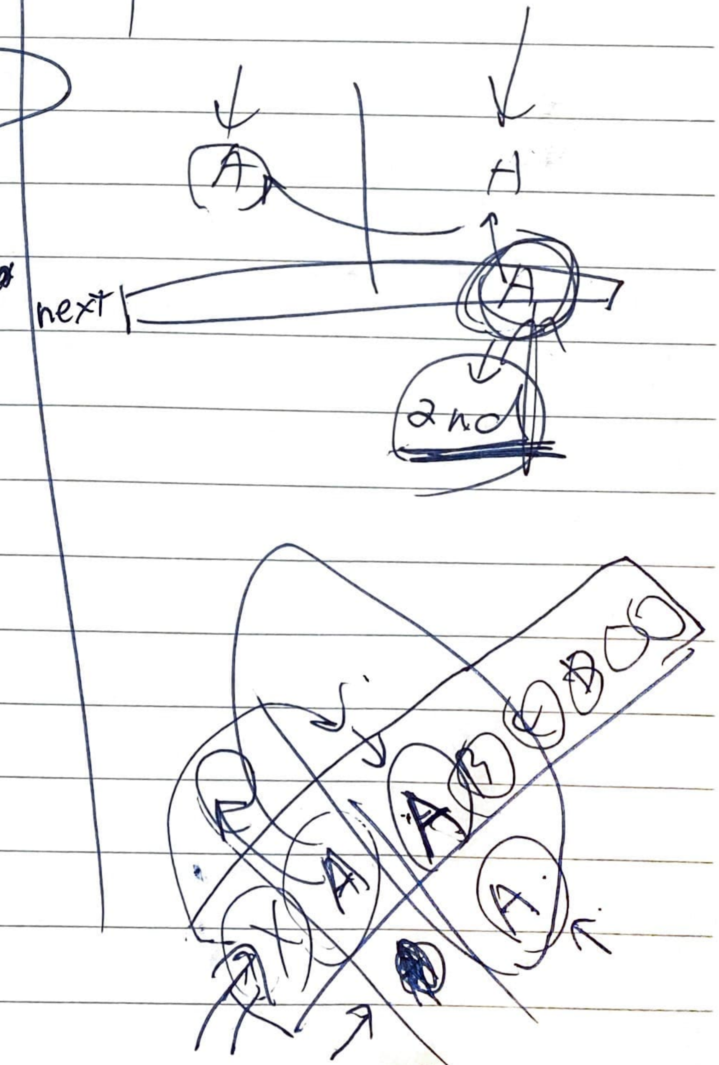
test_current

~~study_next~~

test_next ✓

~~study_prior~~

test_prior.



有variable

来源于上一个 → stored之前

每个list = [固定的值] [缩写]

继承下一个

pre-assign

更新

具体例子程序中：

"f" "ba" "r"

1, 2, i=0 Study Test 10.0, 93.

i=0 all tested. 全test

Study test

all tested only not last.

15

i=8 test 5 15

i=9 only not last 1, 2. 3. 5

① 在 study/test 上面储存每个id的 position.

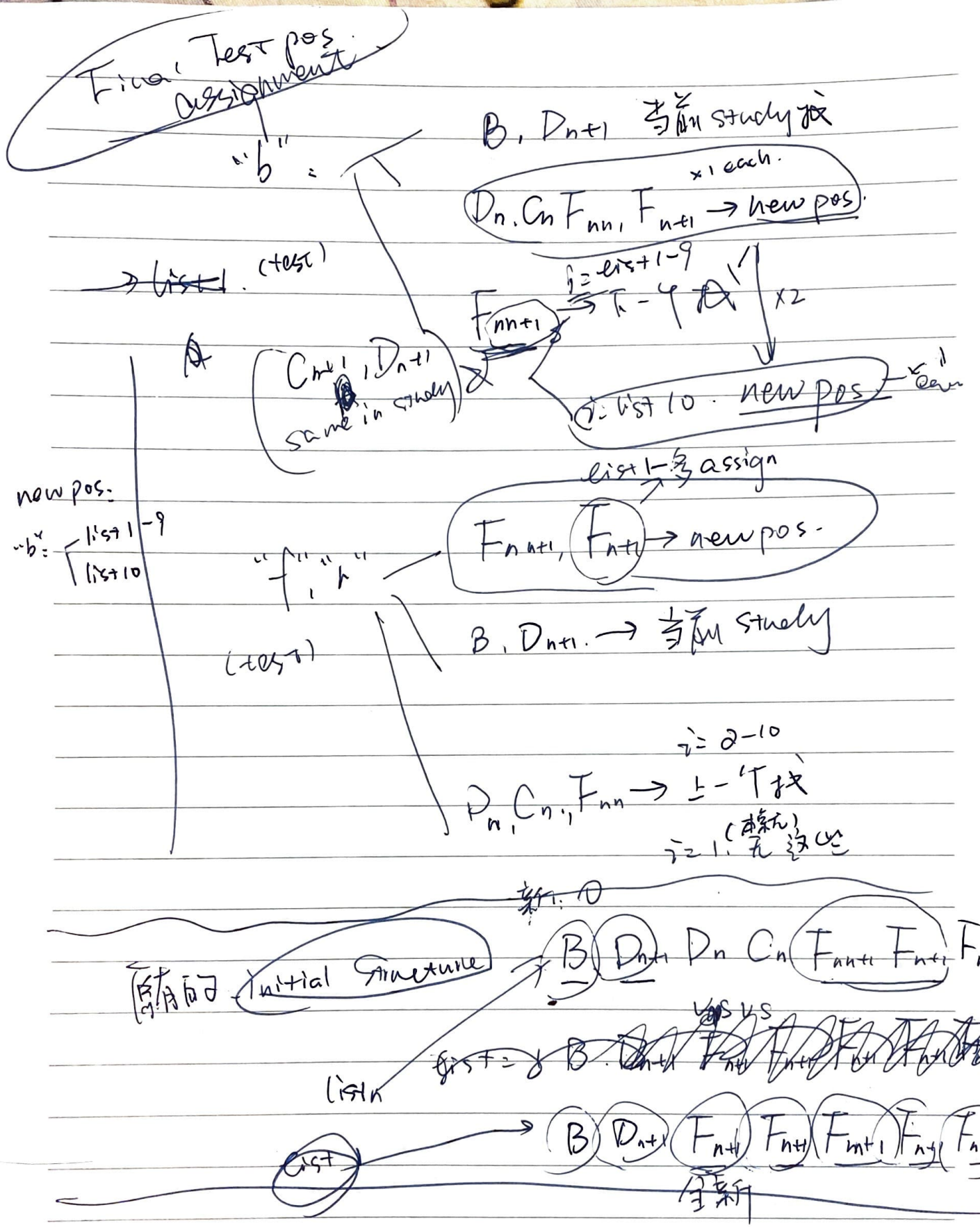
id.

② store 一个给下一个用

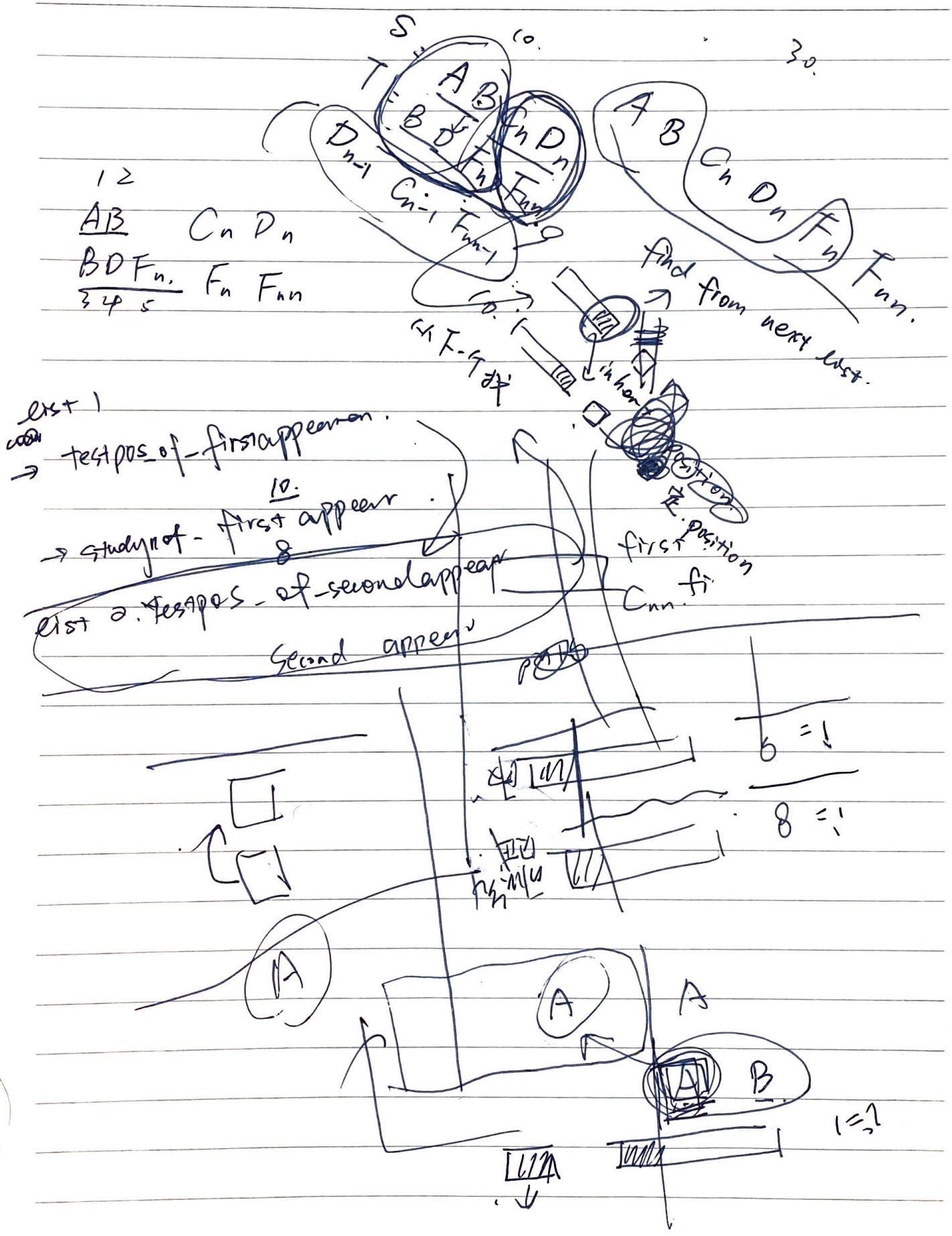
(从上一个获取)

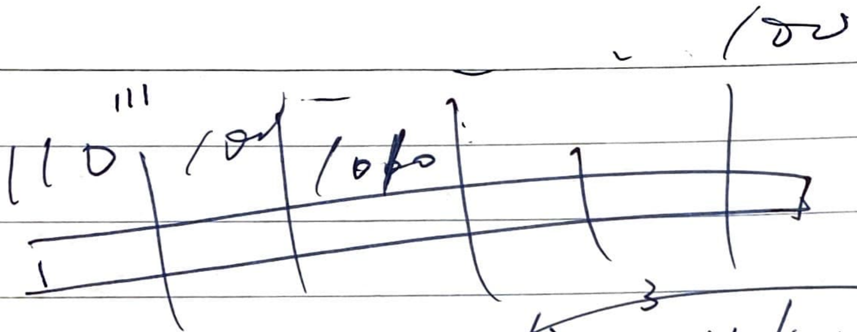
8

9



Final Test showed ×2 whenever new positions


find from next last.
first position
second appear
A
B



item in unit / unit size = 100.

110. a–109 i=0. 0×[ ] i=1 us=4
[0, 109]

110–209 i=1 [409,] +100 us=3
us=4

210–309 [ ] us=3
100+2100. +300

15

15×3=45

item in unit 是 询问 45×10×2=900 ×2/3 =600

一共 10+5=15

480+8

12×2/3=8
4

12×3×10×2×2/3

8×3×2×10+

=36×10×2×2/3

=

360
2
720
24

=720×2/3
=480 +

Total Test in Final T

study 1-n

test 1-n.

array

array [[obj], [obj],]

1. Finer Test # 定condition 的方法.

2.

Q: JS 在map里能用 recursive 计算吗?

A: 4×10 list = 40个 ×3 = 120

B: 4×10 list = 40个 ×3 = 120

3个计算方法

D_{n+1} : 1×10×3 = 30.

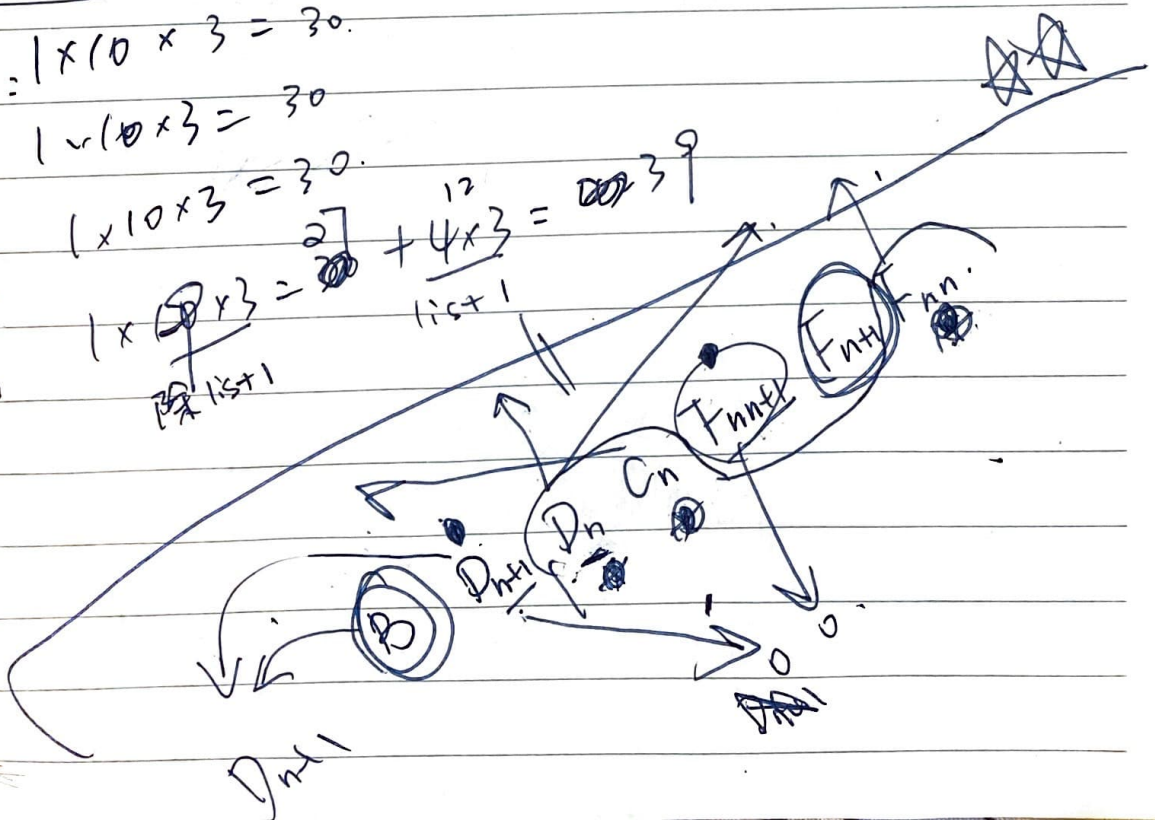
C_{n+1} 1×10×3 = 30

F_{nn} 1×10×3 = 30.

F_n 1×[illegible]×3 = [illegible] + 4×3 = [illegible]

[illegible] list+1

list+1



绘画论
→ 情绪 (心)、经验验证
→ 目的性.

[A A A A C_n B]

Array → 的类型 只保留 A . B . C_{n+1} P_{n+1} F_{nn+1}

↓
最后一个完成的
(F_n, D_n, C_n)

for (
→ { 目的 做出每个 list 里面的 object of all (limited) kind.

→ array of array 不同类型的个数 - 4
同样的第一层 dimension 数量 of
枚举 merge subdimension of array
each array

from each list	from last list
ifA	ifD_{n-1}
ifB	ifC_{n-1}
ifC_n	ifF_{nn-1}
ifD_n
ifF_{nn}
ifF_n