11 3 1

1 x 1 + 1 x 2 + 1 x 3 = 6.

3 x 1 + 1 x 2 + 1 x 3 = 8

+ 3 x 2 + 1 x 3 = 10

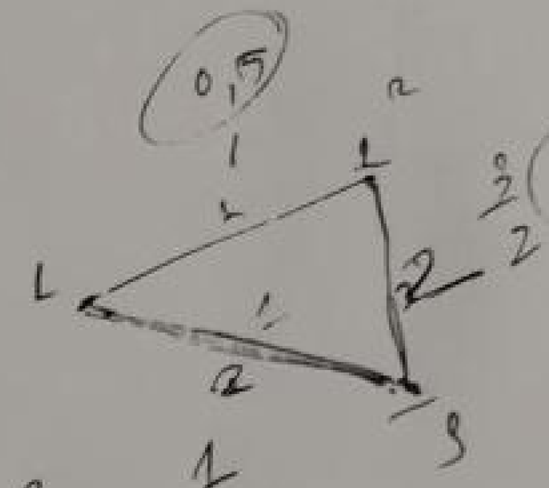
1 x 1 + 3 x 2 + 1 x 3 = 12

9 x 1 + 9 x 1

N=4 khoá 2 gọi 2

$\overset{K\ K}{\sqcup\sqcup\sqcup\sqcup}$ $\overset{(n-3)}{K.K-1} + N + \frac{1}{KKKK}$

2.



4 điểm

$1 \xrightarrow{2} 2 \xrightarrow{3} 3$

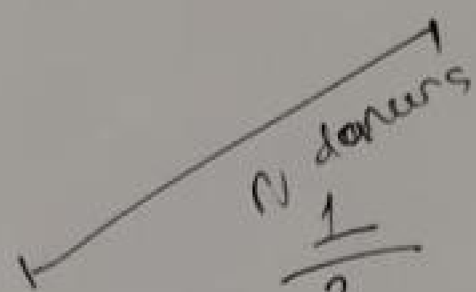
$1 \to 2 \to 1 \to 3$

$1 \xrightarrow{2} 3$

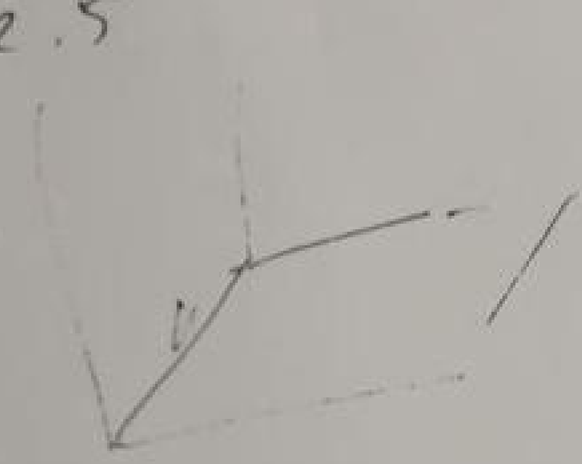
2 điểm

2 $\frac{2}{4}$ 1 2

$\frac{1}{1} + \frac{3}{2} = 2.5$

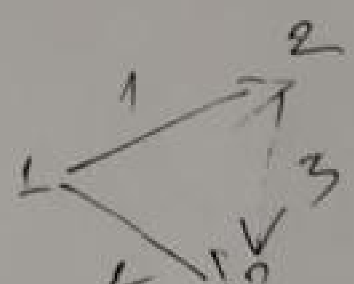


$\frac{1}{2}$



$\frac{1}{2}$

2.57



$1 \to 2 \to 3$

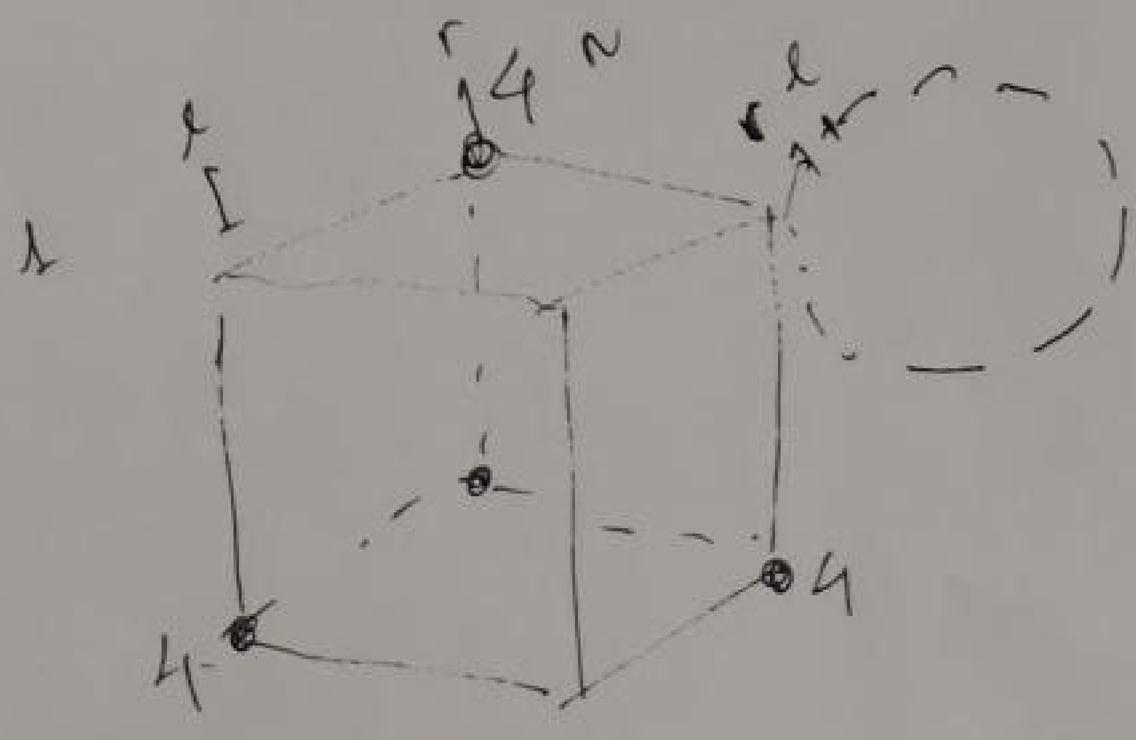
$1 \to 3$

$E(1) = 0$

$E(12) = \frac{1}{2} \cdot 1 = 0,5$

$E(123) = \frac{1}{2}(0.5 + 3) = 1.75$

$E(13) = \frac{1}{2} \cdot 2 = 1$



1/1.75

n and K.

$k \leq n-1$

$n = 4 \rightarrow K = [0, 1, 2, 3, 4]$

sort.

(catA_10)_10

→ sort → [catA 10 catA 3 catA 4.]

temp-dat: catA_10

3 8

10 11.

visited.add

catA 3
catA 4
catA 10
catB 1
catB 2
catB 2
catC 5.

NA in W

cat, code = input.split()

if cat not in dct.

dct(cat) + code = code.

if cat + code not in the counter dct:

counter dct [cat + cde] + = 1

sorted. dct

for key in dct

if key split.[0] not in visited.

temp-dct.append [key] = key split [1]

compare case - insensitive.

add a tuple.

a < b

c ≤ d

if a < b

and a < c

b < d

c < d

left < right.

top < bottom

4

bon search.

0.0 3,1 -2,4

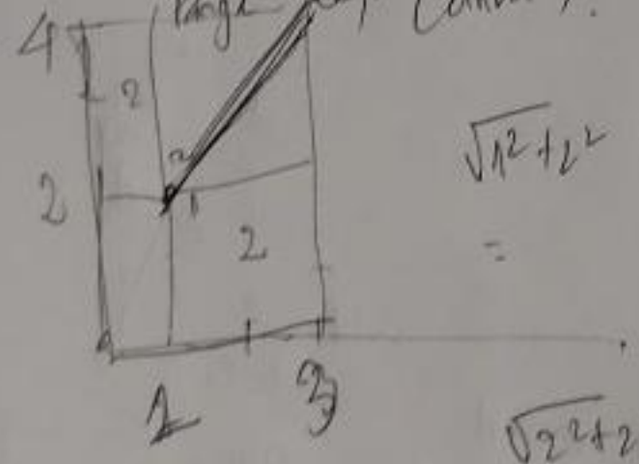
5.0 -2,3

~~abs(ax1 - x0)~~

abs $(x_1 - x_2)^2 + (y_1 - y_2)^2$

max ((contri[0][0] - contri[1][0]) ^{2} + [00

$(a-b)^2$



$\sqrt{1^2+2^2}$ + $\sqrt{2^2+1^2}$

=

$\sqrt{2^2+2^2}$

2

len % k != 0
    return -1
else:
    for row in list.

len = 4

0 % . 0 1 2 3 3
1 3 5 2 0 1 2

xor.   1100.
      ^ 1100.

1 %.   1 % 1 =   is odd.

. % 2 1 = 1

4 0 2 3
0 1

65 11
4 0 0 0
4 3 0 0
0 0 0 0

a[1] = b[1] = [1].

101
010
111   2-1

001   101 0
101   010 1
101   cnt(1) = 1
       cnt(0) = 2
010
111   1 - 2 = -1 % 1 = 1

24
1427 5368
1427 8142

4, 1, 2, 3

0 ≤ ai ≤ 20.

list [

Badgui.ph
Sacha Ps

9

i=0 , i=1 .

right, right

left , +1

right -1

left +1

right -1 .

for a in pairs for log

for pair in pairs :

for left right in pair :

dct . append (lef, +1)

dct . append (right, 0-1)

(-2 ; 1)

for ~~pair in pairs~~

for item in list.

c += item[1]

if c > best[0]

best[0] = c

best[0] = retarn best[0]

b[0] =

0 0 0 0

56 11

2 << 1100 1100 .

101 010.

[0 : 4][4 ; 8]

[i : i + k] for i in

range (0, len(bina), #)

b ≠0

1

for 2 1

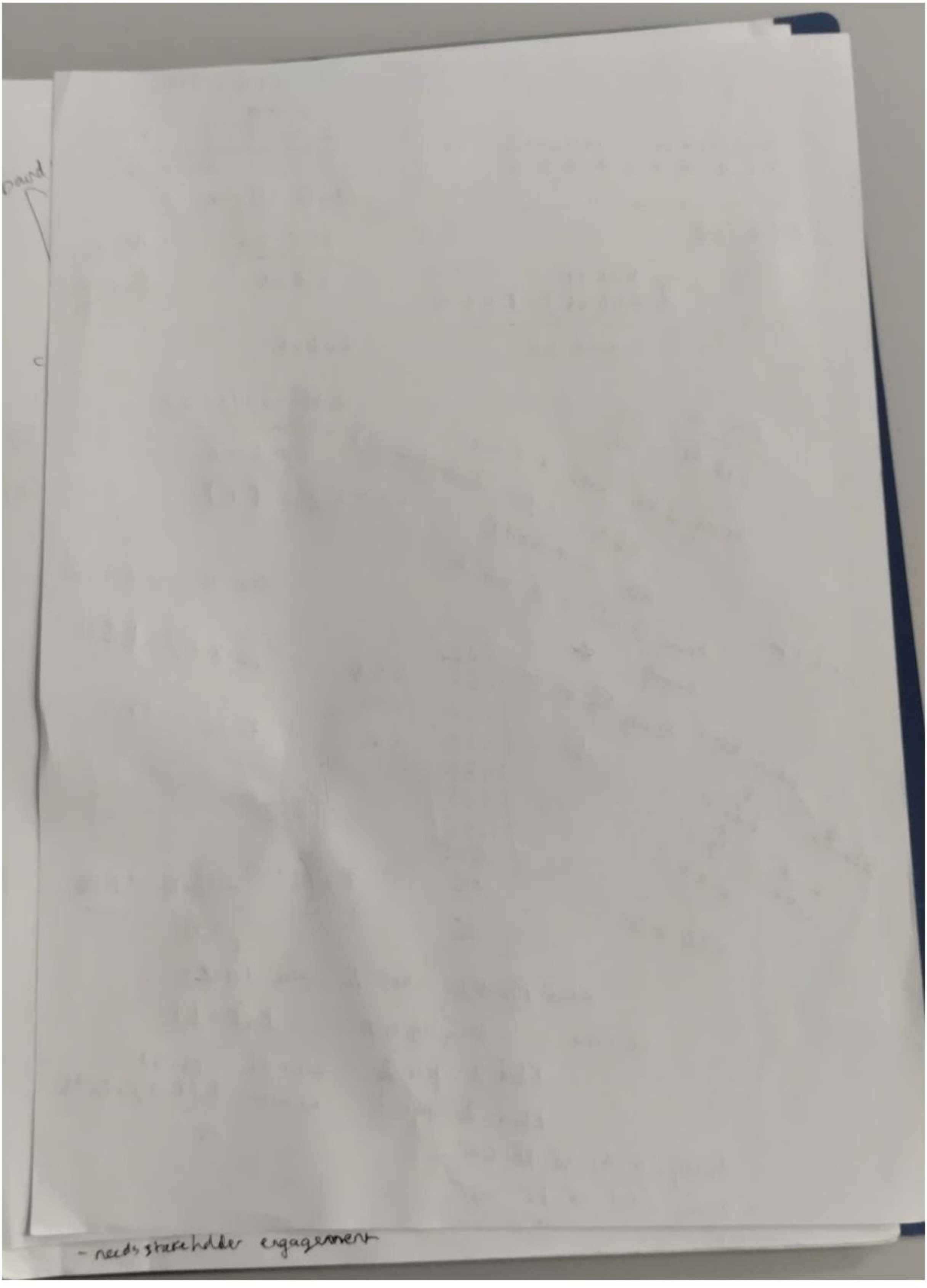
pa.

0 xor 0 = 0

1 xor 0 = 1

1 xor 1 =

count(1) - count(0)


- needs stakeholder engagement