11 3 1

1 × 1 + 1 × 2 + 1 × 3 = 6.

3 × 1 + 1 × 2 + 1 × 3 = 8

1 × 1 + 3 × 2 + 1 × 3 = 10

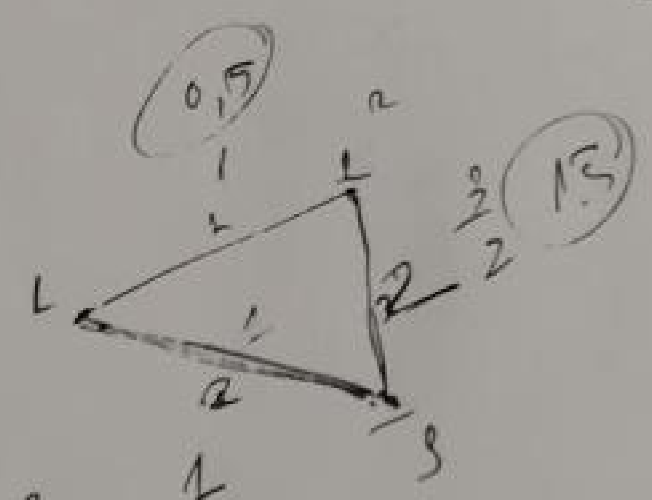
5 × 1 + 1 × 1 + 1 × 3 = 11

N=4 khoá 2 joa 2

K K

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

2.



0,5
1
2
1
2
1
1
2
2
3/2
1.5
2
3

4 điểm

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

$1 \rightarrow 2 \rightarrow 1 \rightarrow 3$

2 điểm

$1 \xrightarrow{2} 3$

1

2

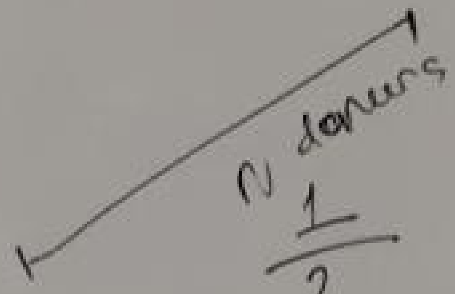
2/4

4

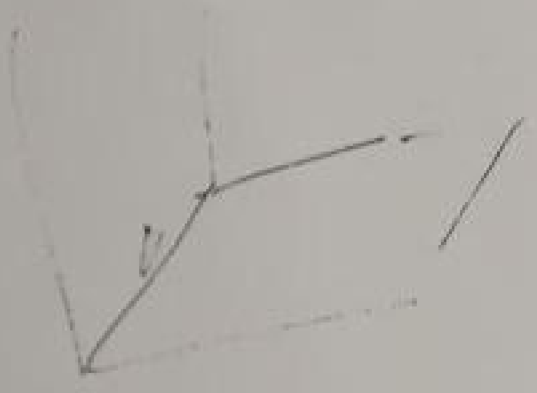
1

2

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



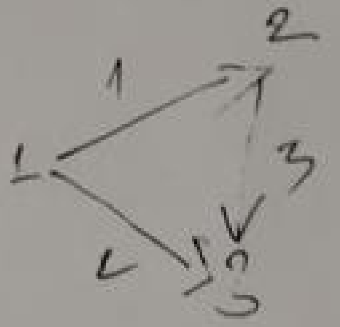
$\frac{1}{2}$



1

2

2.5+



$1 \rightarrow 2 \rightarrow 3$

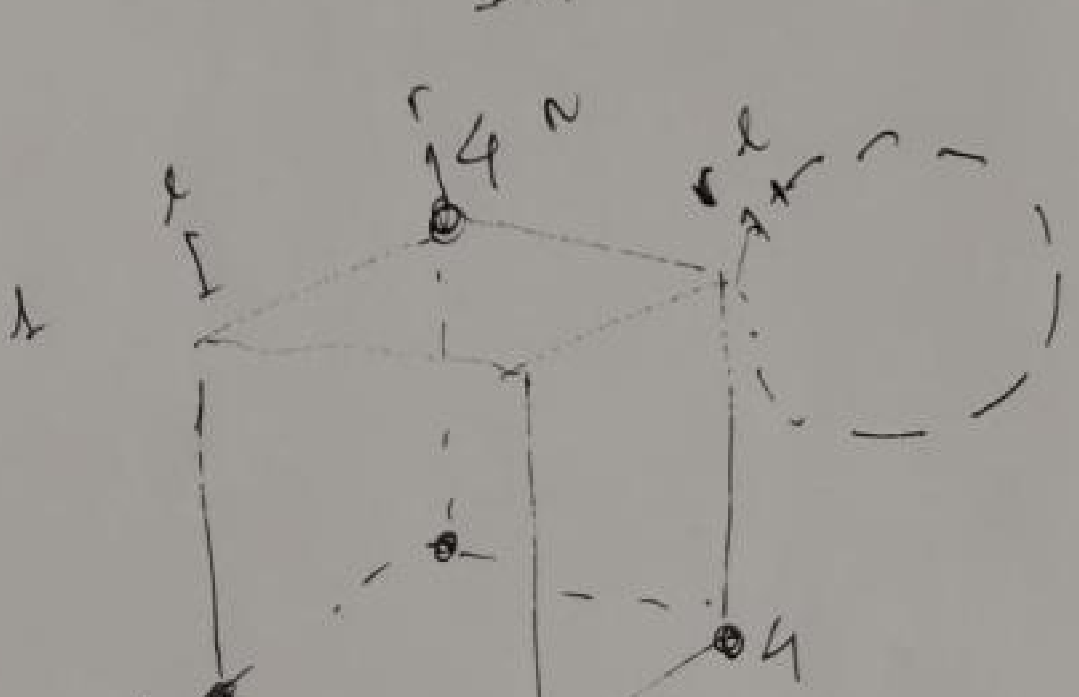
$1 \rightarrow 3$

$E(1) = 0$

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

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

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



1/1.75

1,75

n and k.   n = 4 -> k = [0, 1, 2, 3, ~~4~~.

k ≤ n-1

sort.

[catA_10]_10

-> sort -> [catA 10   catA 3   catA 4.]

temp-dct: catA_10

3 8

10 4.

visited.add

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

not in v

```
cat, code = input.split()
  if cat not in dct.
    dct[cat + code] = code.
  if cat + _ + code not in ~~xx~~ count~~dct~~ dct:
    countdct [cat + code] += 1

sorted. dct
    for key in dct
        if key split.[0] not in visited.
        temp-dct ~~append~~ [key] = ~~key split[1]~~
```

a < b
c < d

if a < b
and a < c

b < d
c < d

left < right
top < bottom

compare case-insensitive

add a tuple.

4

bin search

0,0 3,1 -2,4

5,0 -2,3

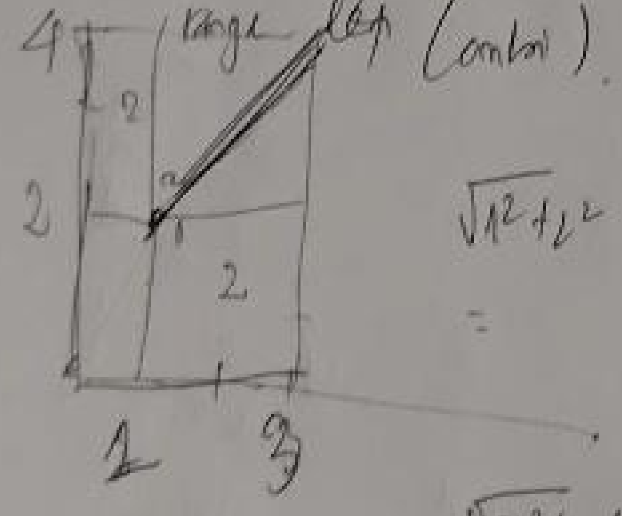
~~abs(ax_1 − x_0)^2 +~~

~~abstan~~ $(x_1-x_2)^2+(y_1-y_2)^2$

max ((contr[0][0] - contr[1][0])^2 + [00

$(a-b)^2$

range len (contr).



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

=

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

2

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

len = 1

3 3

0 % . 0 1 2 5 7 0 1 2

1 3

xor. 1100.
^ 1100.

is odd.

1 %. 1 % 1 =

. % 2 1 = 1

4 1 2 3
0 1 2

6 5 1 1
4 0 0 0
① 4 3 0 0
② 0 0 0 0

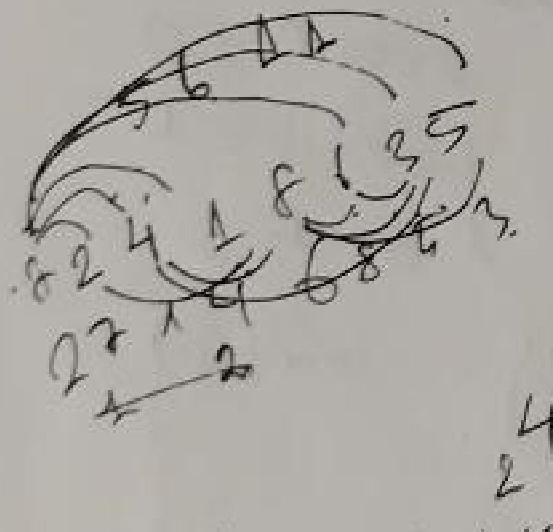
a[1] = b[1] = [1) .

1 0 1
0 1 0
1 1 1

2 - 1

1 0 1 0
0 1 0 1

0 0 d
1 0 1
1 0 1
0 1 0
1 1 1


5 6 1 1
3 5
8 2 4 1 8 1
6 8 6 3
2 7 1 4
1 2

can (1) = 1
con (0) = 2

1 - 1 = -1 % 1 = 1

4, 1, 2, 3

1427 $^{2^4}$
5368
1427 8142

list [$0 \le a_i \le 20$.

bodyq.ph
sucha ps

i=0 i=1.

```
left, righ      left , +1
                righ -1
                left +1
                righ -1 .
```

1.≪ 1100 110 0.

```
for a in pairs for lef

for pair in pars:
    for left right in pair :
        dct . append ( lef, +1 )
        dct . append ( righ ; 0 -1 )

(-2 ; 1)

for ~~pair in pars~~
    for item in list
        ct = item[1]
        if c > best[0]
b[0] =      best[0] = c
    return best[0]
```

0 0 0 0

5 6 1 1

101 010.

```
[0 : 4][4 ; 8]
[i : i + k] for i in
    range ( 0, len(bina), 4)
```

5 10

1 ~~21~~ 2 1

raw.
0 xor 0 = 0
1 xor 0 = 1
1 xor 1 =

count(1) - count(0)

band

- needs stakeholder engagement