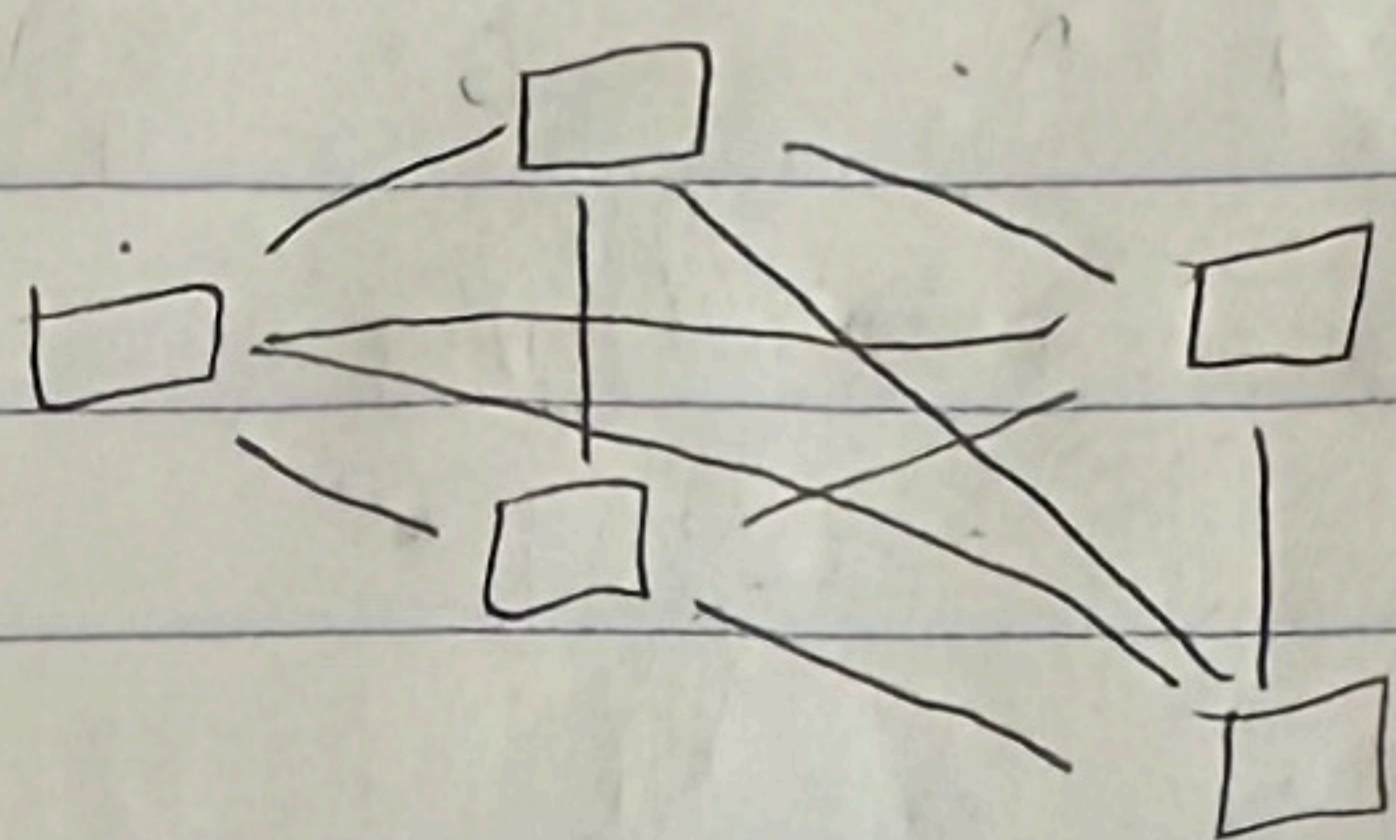


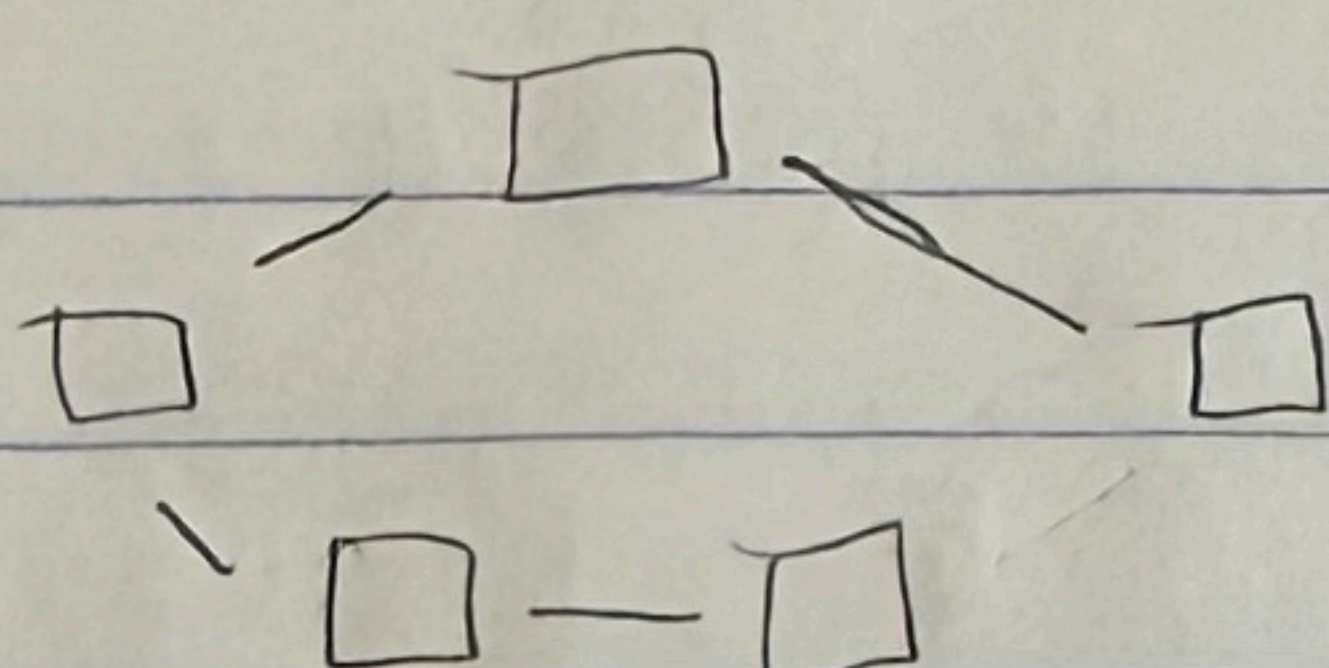
Trees



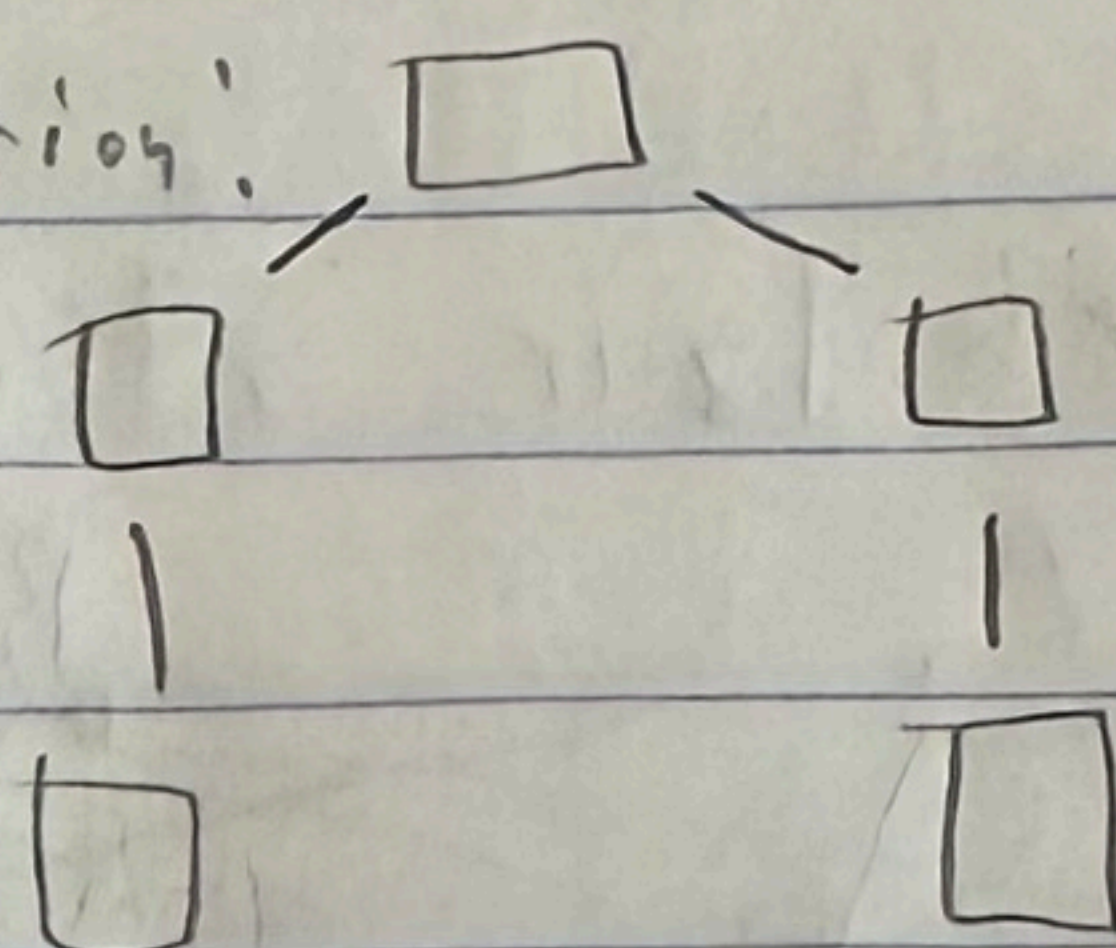
Routers

- connect them with
least amount of wires

One solution:



Another solution:



$N=5$

4 connections

Spanning Tree

- ① Connected to all nodes in graph
 - ② Use fewest possible edges
- Tree has no cycles

N -nodes

- unique path for every a, b

must have $\geq n-1$ connections

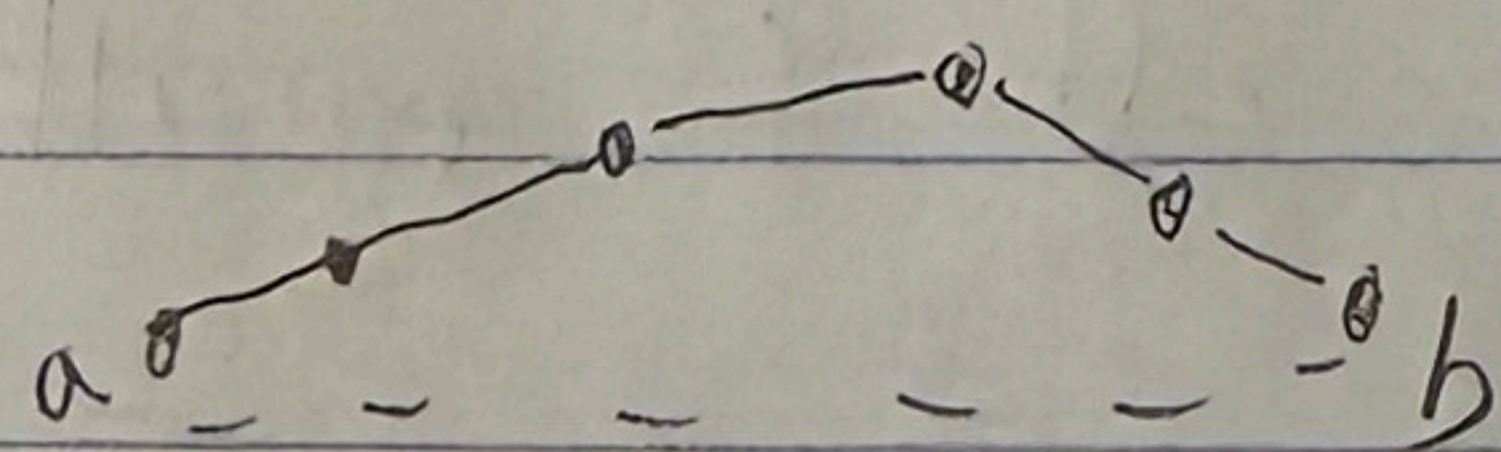
otherwise it will be disconnected

$\leq n-1$

keep removing

edges until

something disconnects.



When you make it a tree,

you are claiming only one way

between nodes.

By contrapositive,

Every path is

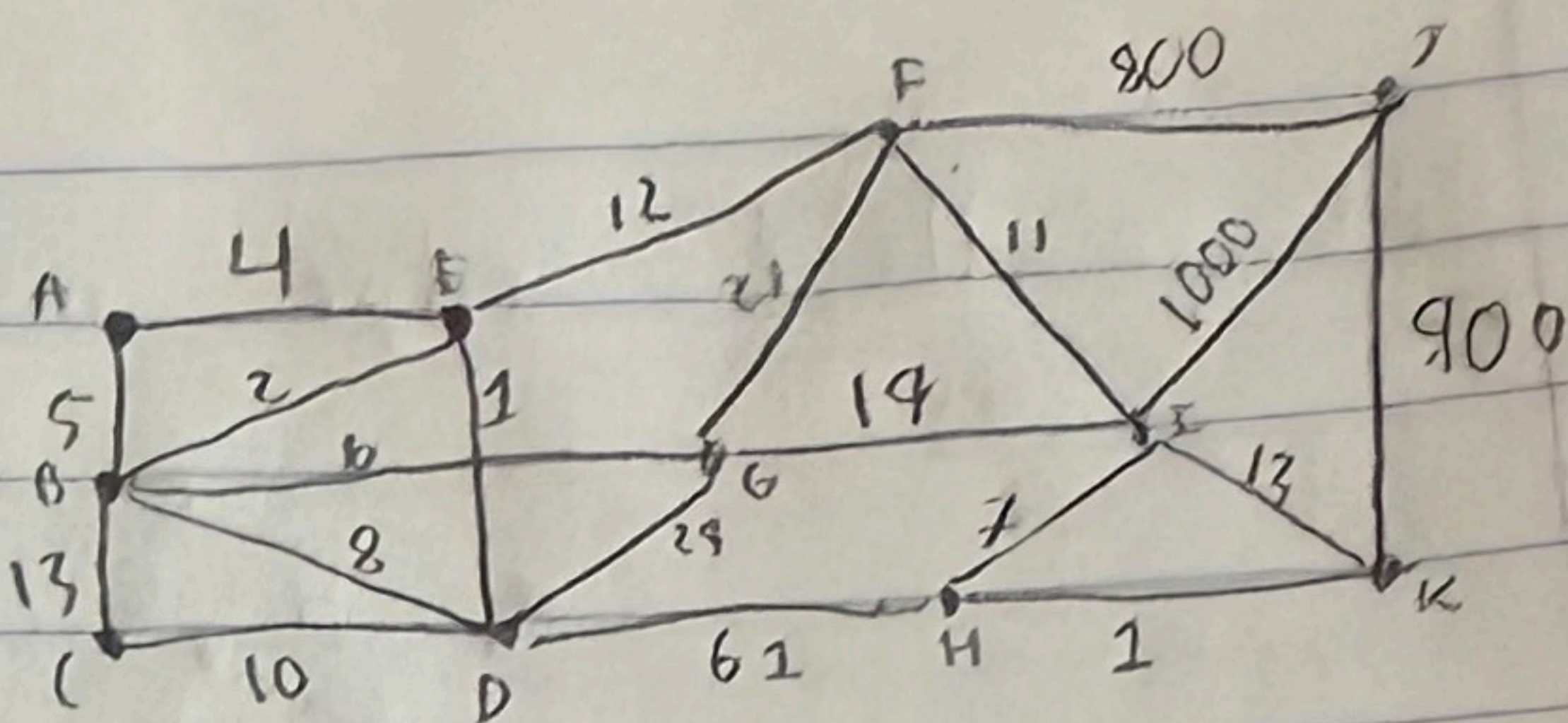
unique, but only

1 path.

Hurricane Problem!

cities bridges

cost
to
fix



Minimum Spanning Tree (MST)

Goal:

- ① All cities are connected
- ② At lowest possible cost

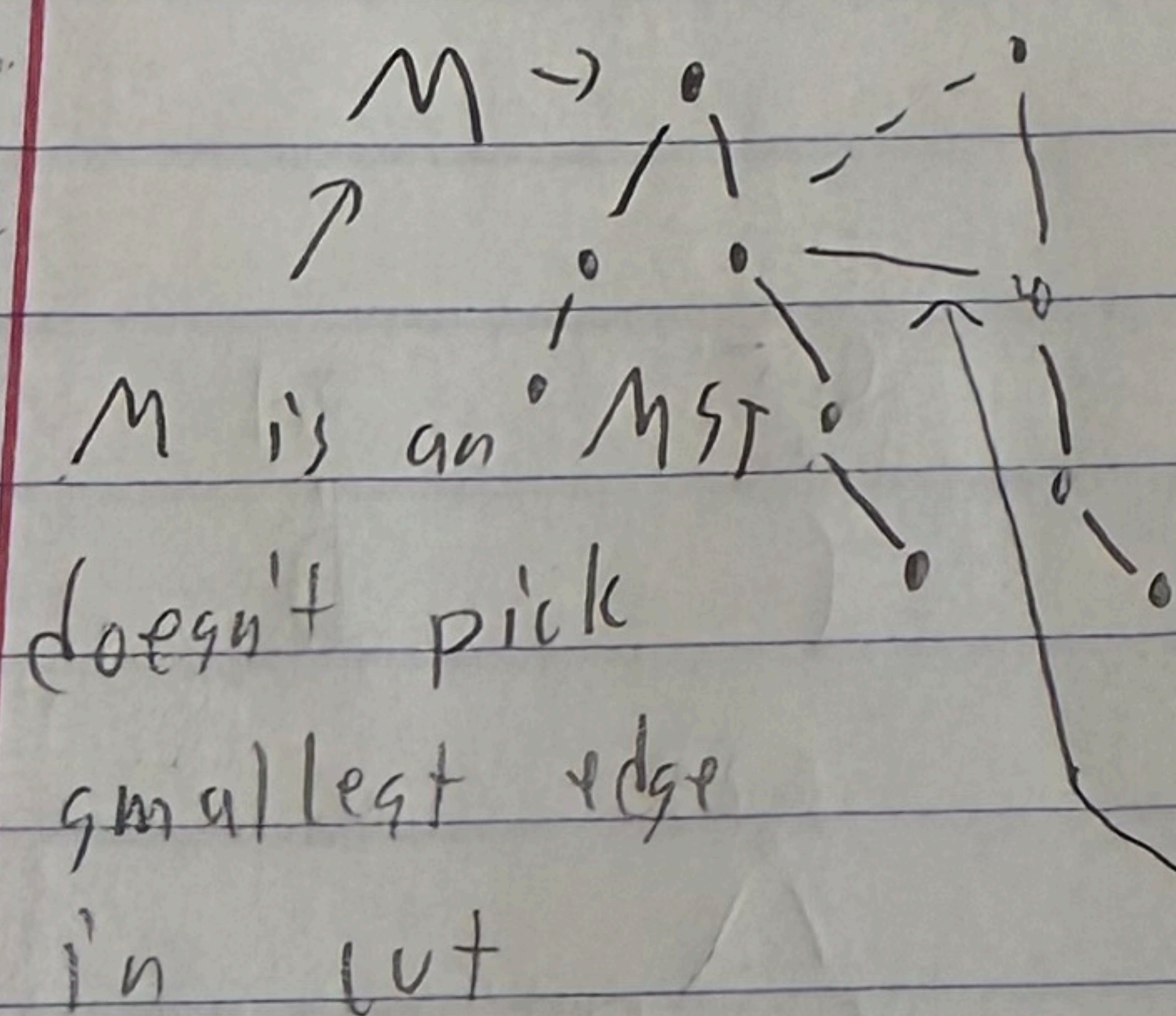
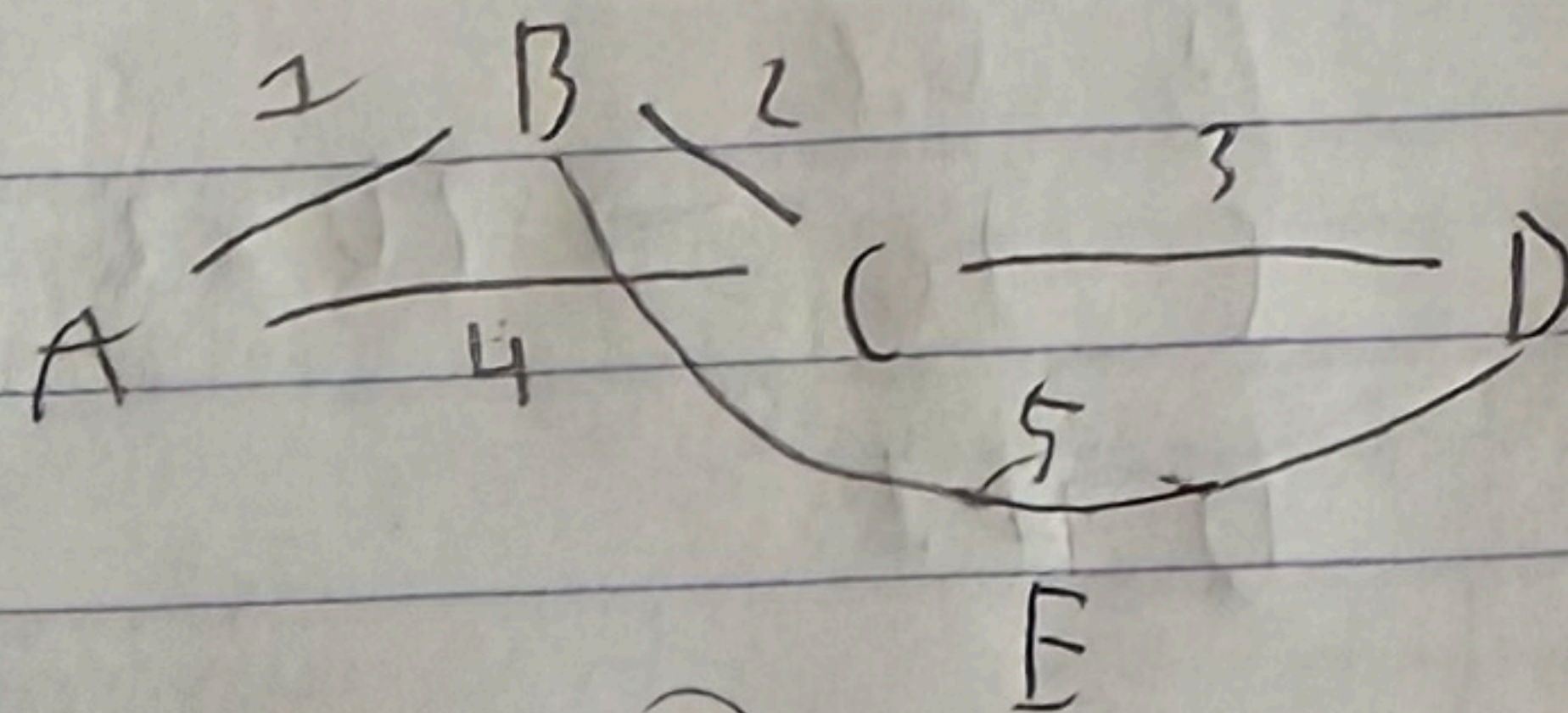
In some cases, you must choose most expensive

If weights are unique there is only one spanning tree

Cut Property:

- Any partition, smallest edge must be in MST

Ex.



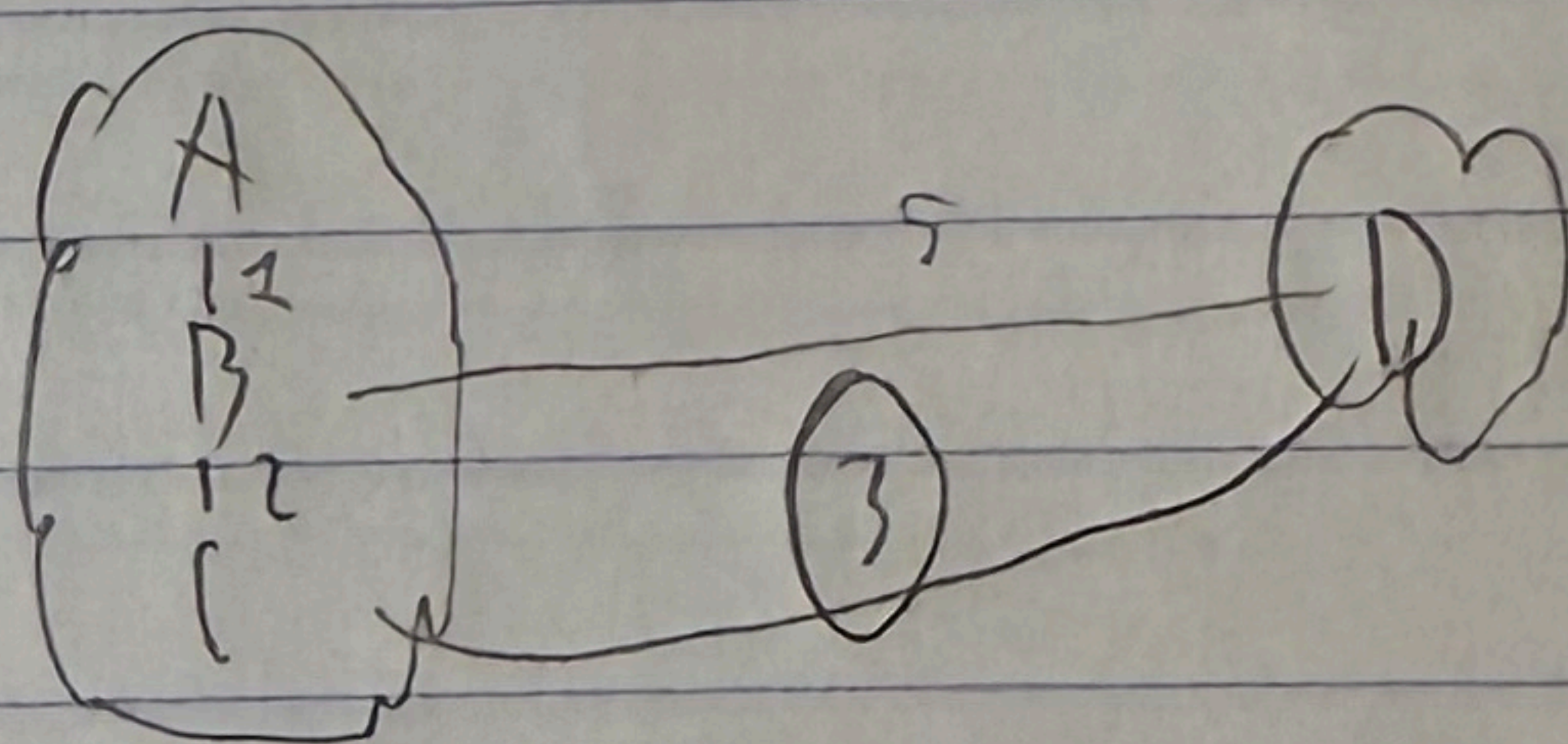
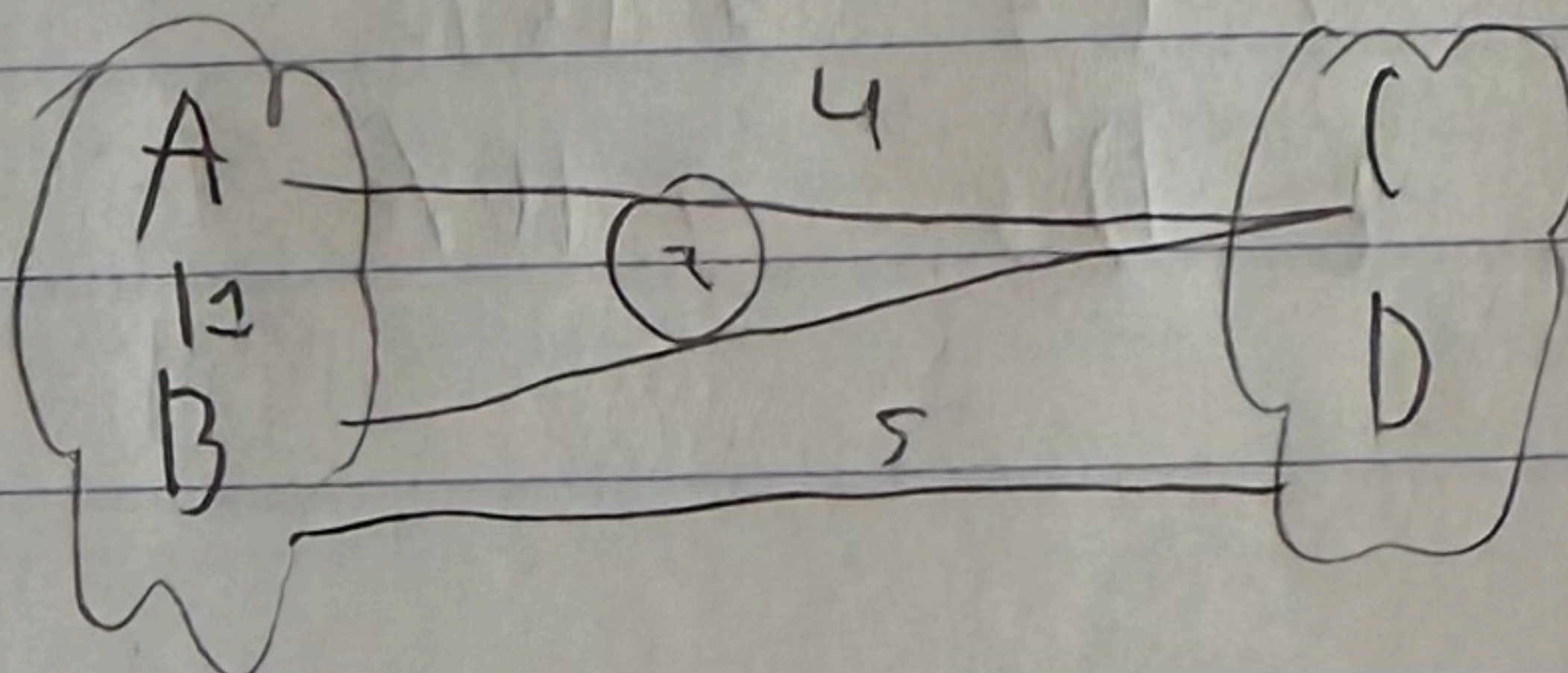
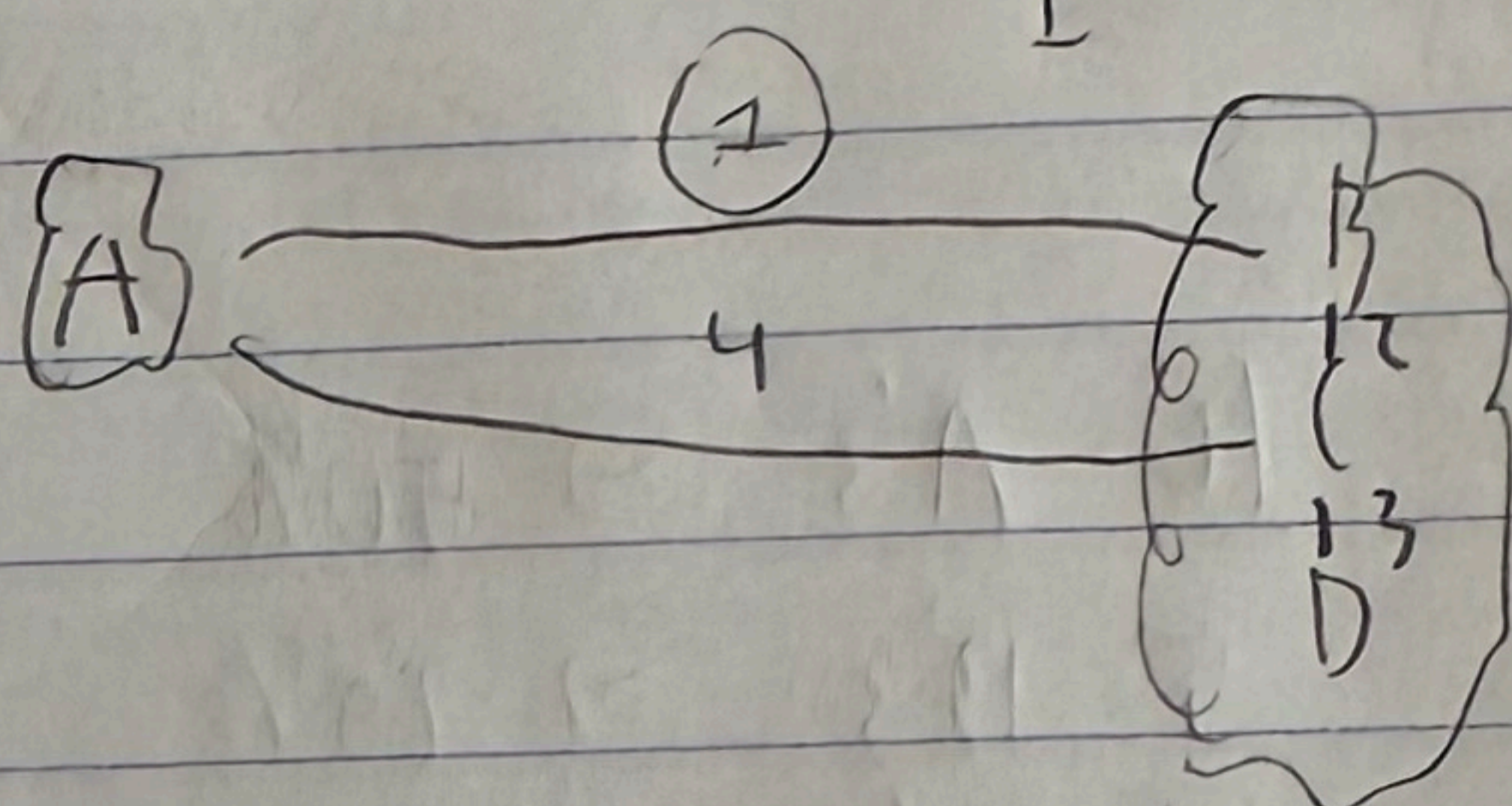
Add → smallest edge

n-nodes

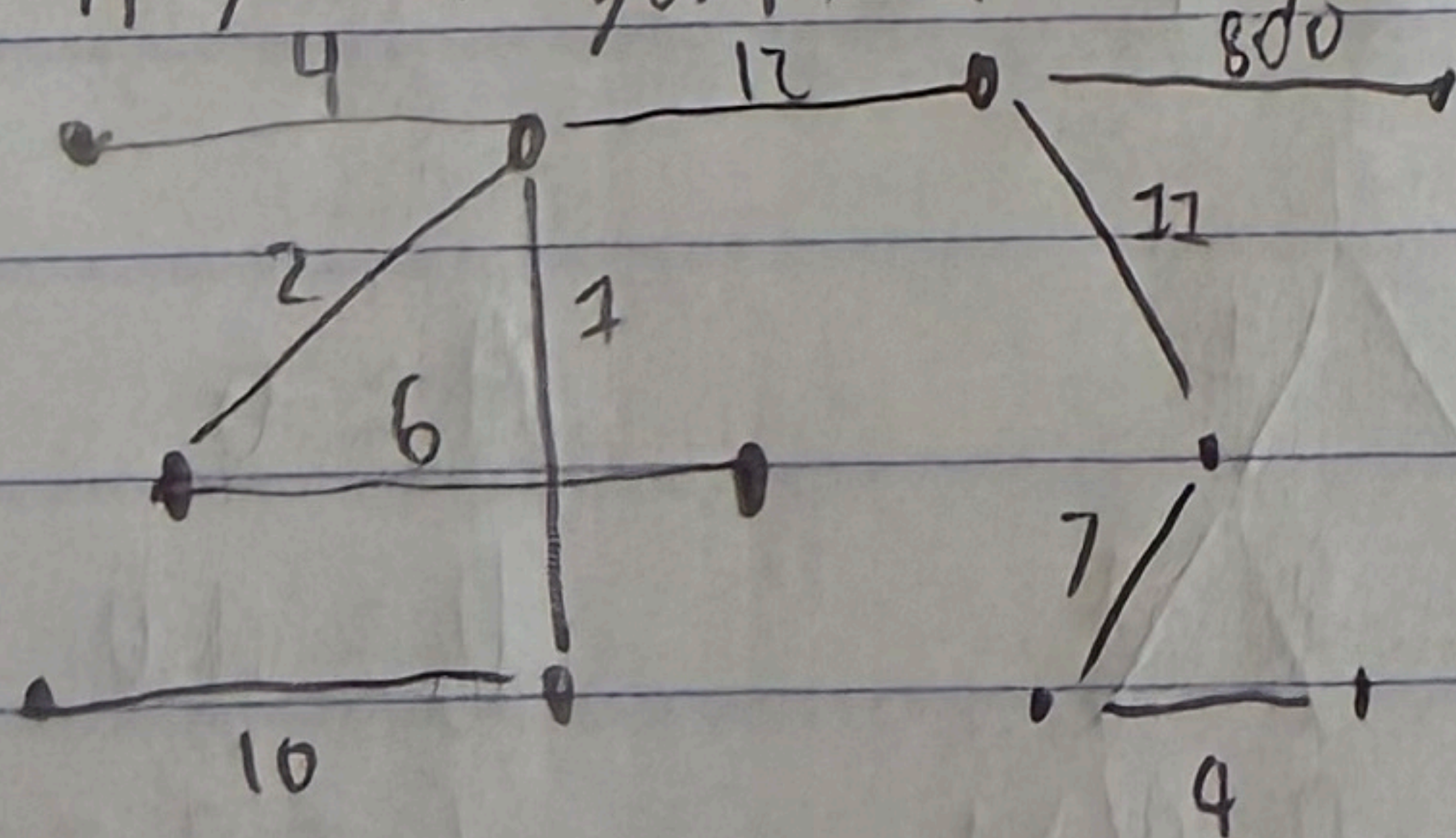
n-edges

↓ Cycle exists

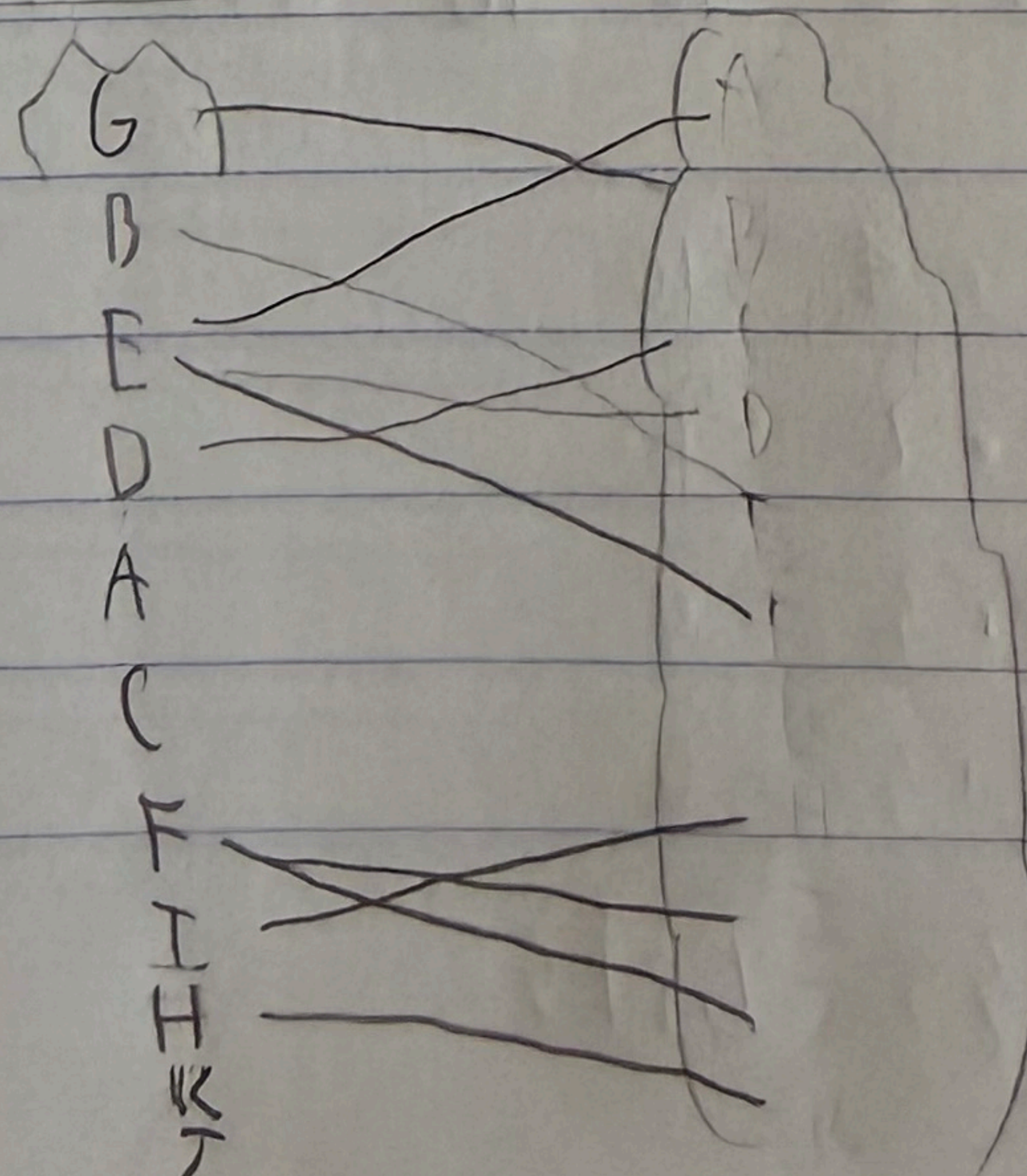
Breaks non-smallest edge - Cheaper Tree



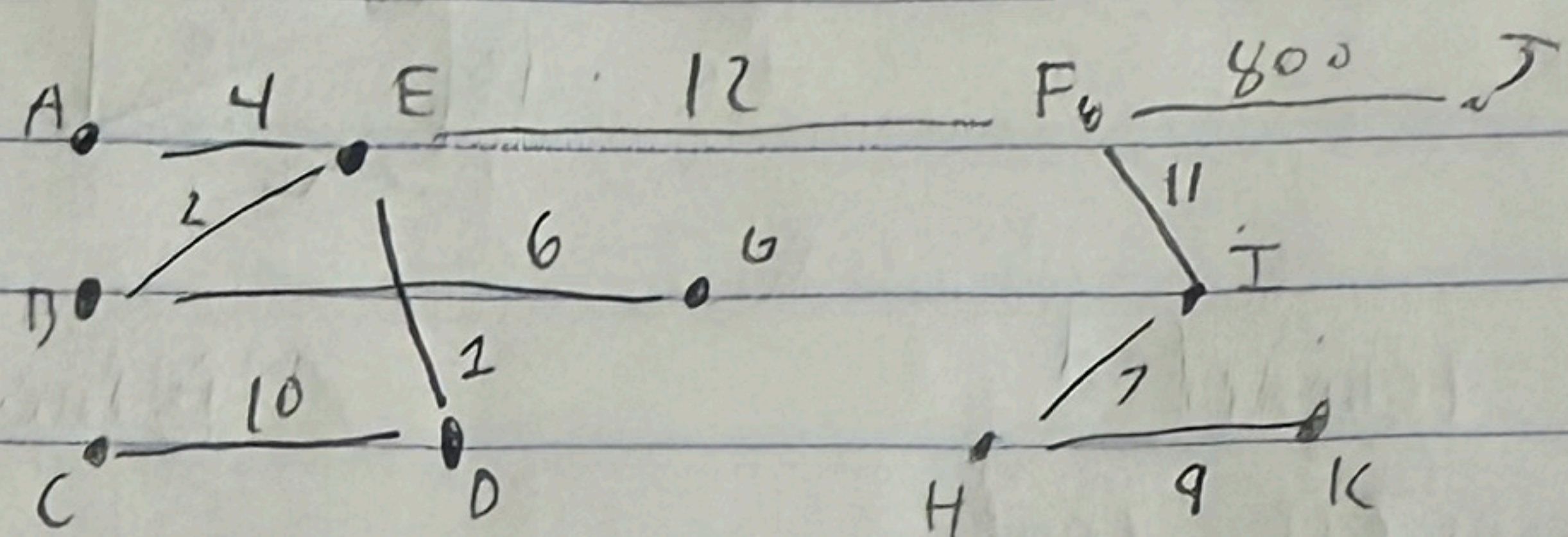
Will's Algorithm - Start At Middle



total: 862



▷ Add Smallest edge one by one, as long as no cycle



Compression

20 characters

AAABBCDAAABCD CBABAAA

A 00 00 00 00 01 ...

B 01

C 10 → 20 × 2 bits → 40 bits

D 11

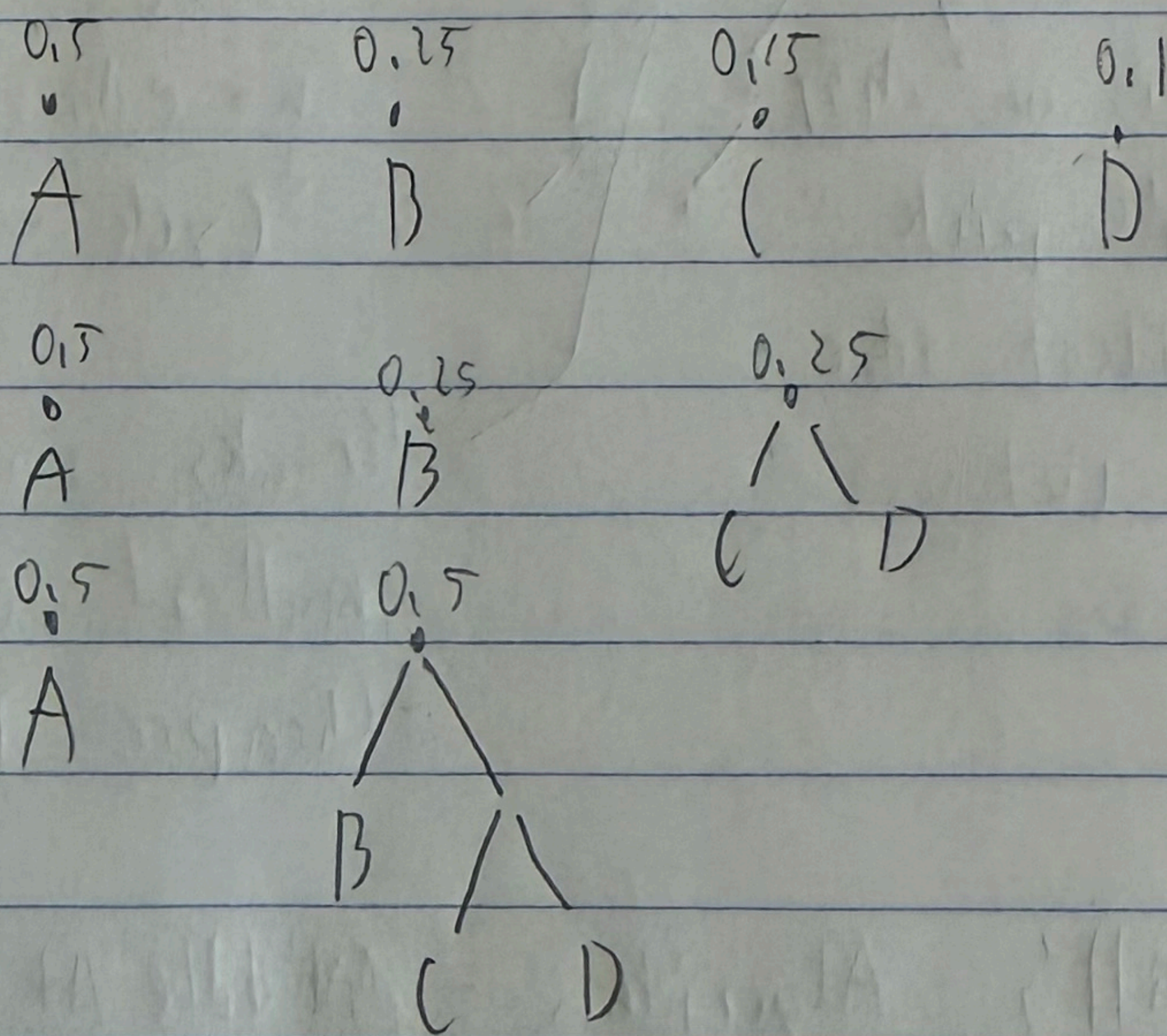
can you do better?

10 A's → 50%

5 B's → 25%

3 C's → 15%

2 D's → 10%



0 0 0 10 10 110 111

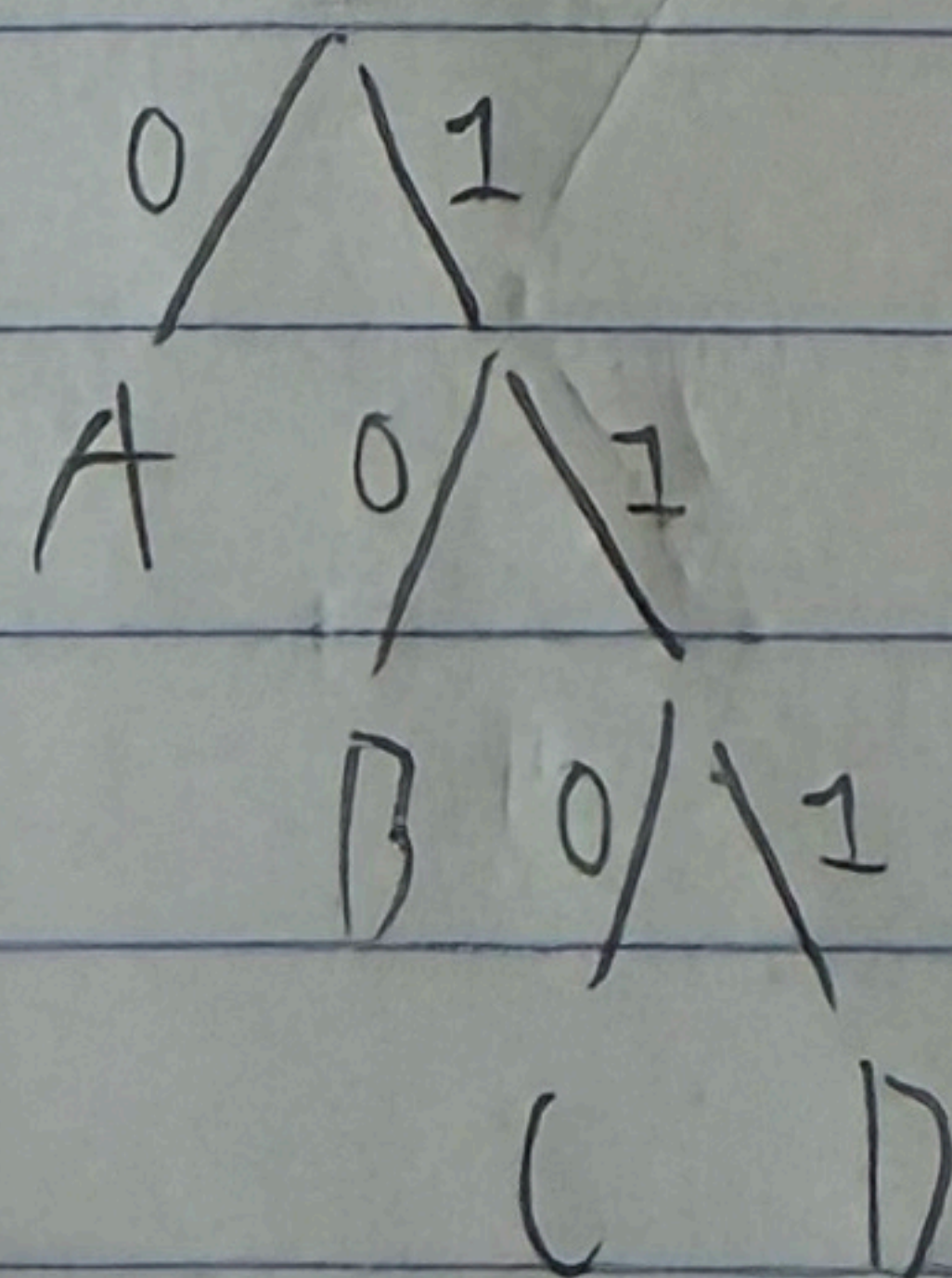
10 × 1 = 10

↑ 5 × 2 = 10

3 × 3 = 9

+ 2 × 3 = 6

= 35 bits



A = 0

B = 10

C = 110

D = 111

$$\begin{aligned}
 &0.5 \cdot 1 = 0.5 \\
 &\uparrow 0.25 \cdot 2 = 0.5 \\
 &0.15 \cdot 3 = 0.45 \\
 &+ 0.1 \cdot 3 = 0.3 \\
 &= 1.75
 \end{aligned}$$

$$1.75 < 2$$

- more efficient