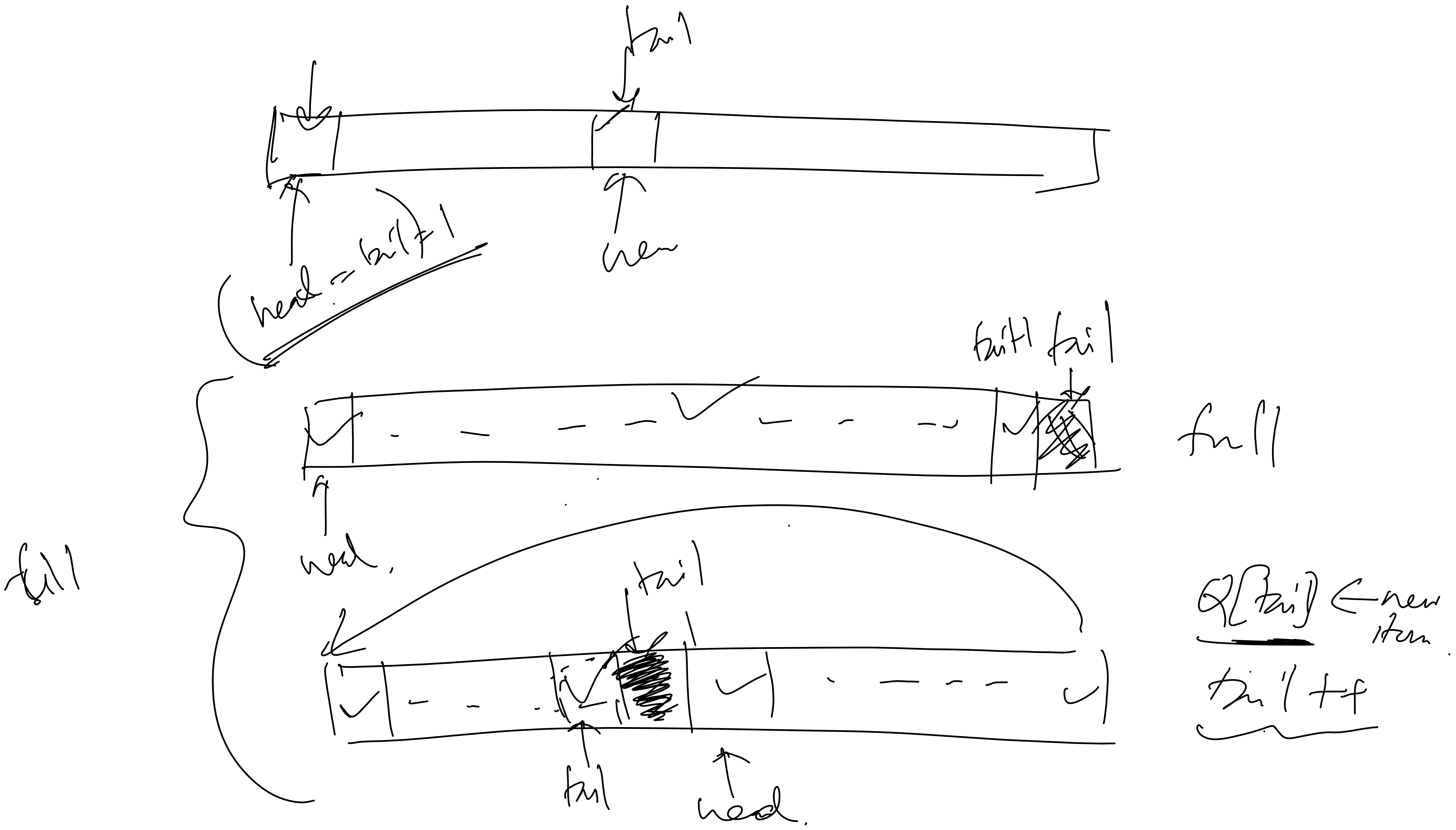
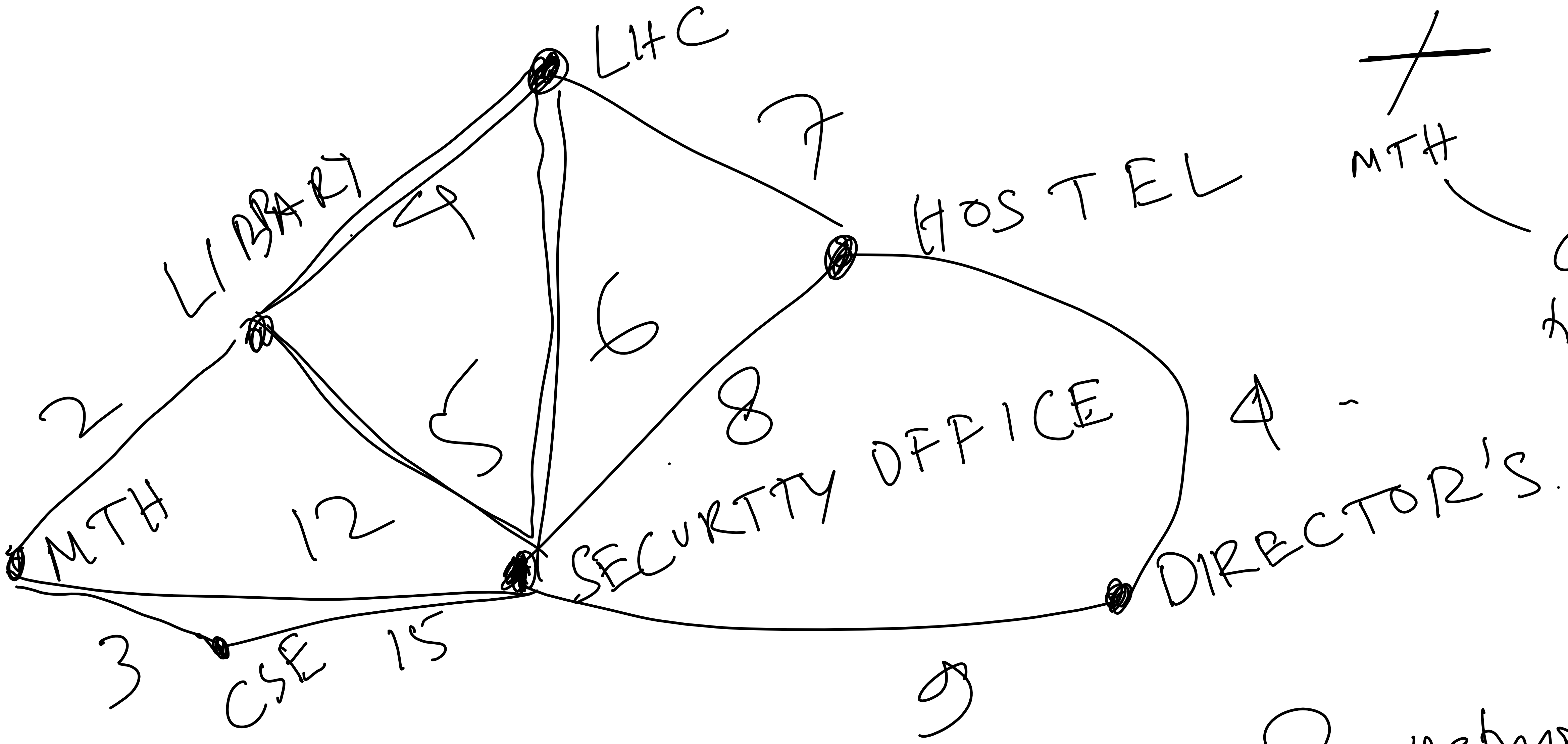
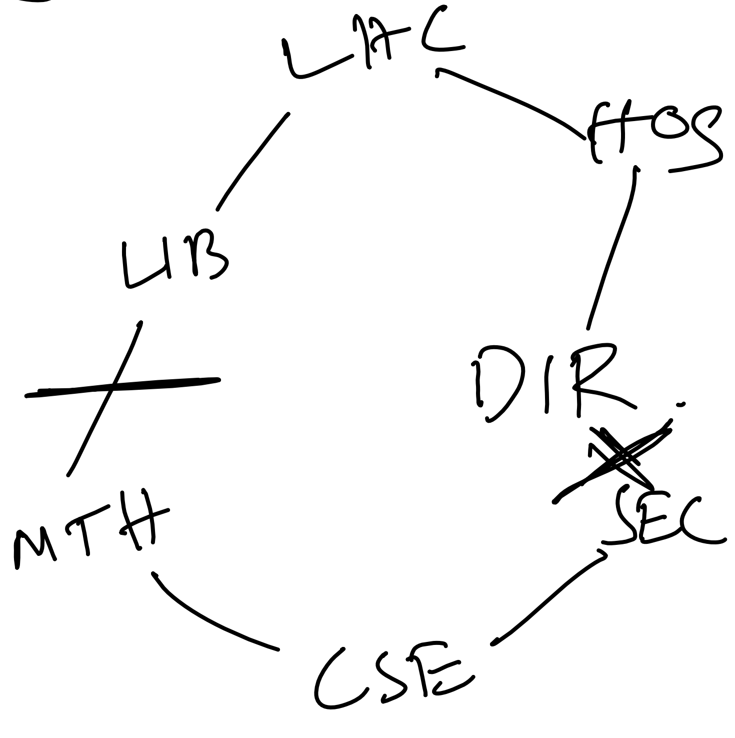

tail
head == tail+1
new
tail+1
tail
full
head
full
tail
tail
head
Q[tail] ← new item.
tail++

Module: Greedy & Dynamic Programming

Greedy algorithm design





tree.

$\binom{n}{2}$

(1) entire network must be connected
(2) cost must be minimized.

} network design

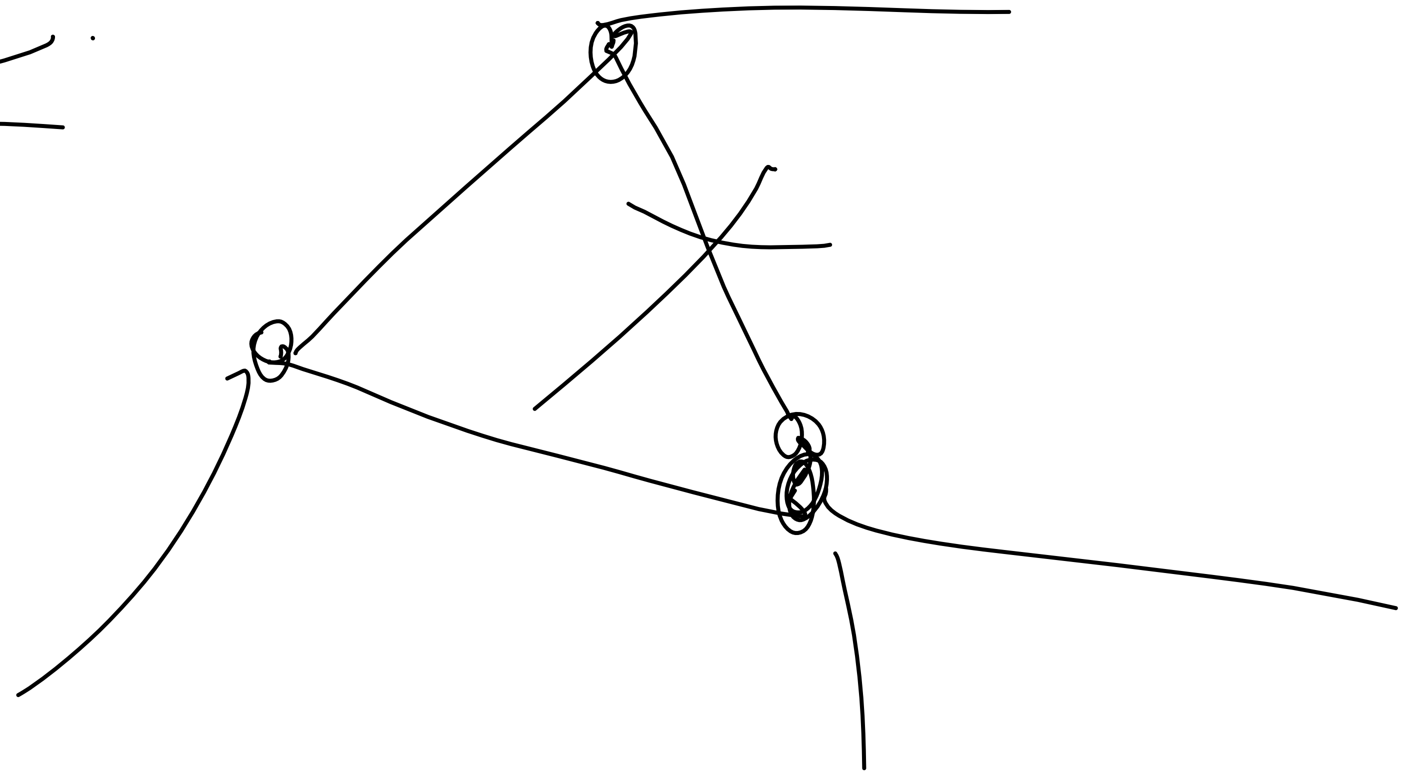
~~Wha~~ only works for "undirected graph"

LHC

HOSTEL

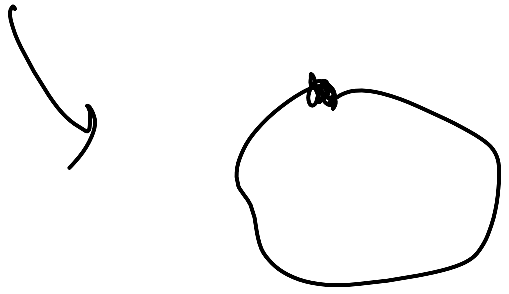
LIBRARY —— LHC — 4 — SEC-5 — MTH-2.

Obs: Cycle in the output graph is never possible.



Def^n [undirected tree]

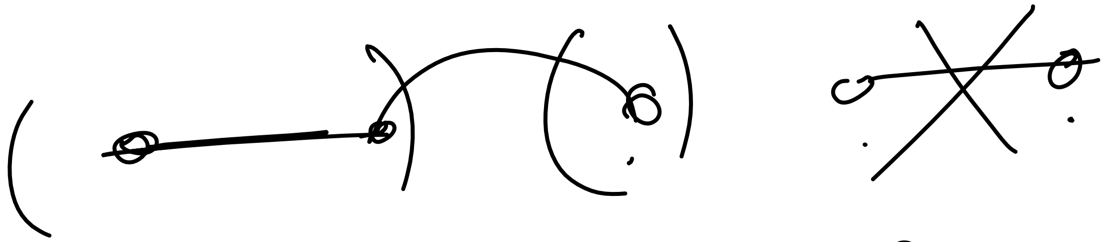
An undirected graph
- connected [$\exists$ path between all vertex pairs]
- acyclic.



Claim: Let T be an undirected tree on n nodes. T must always have $(n-1)$ edges.

Proof:

Hint:



n

~~8~~ . ~~7~~

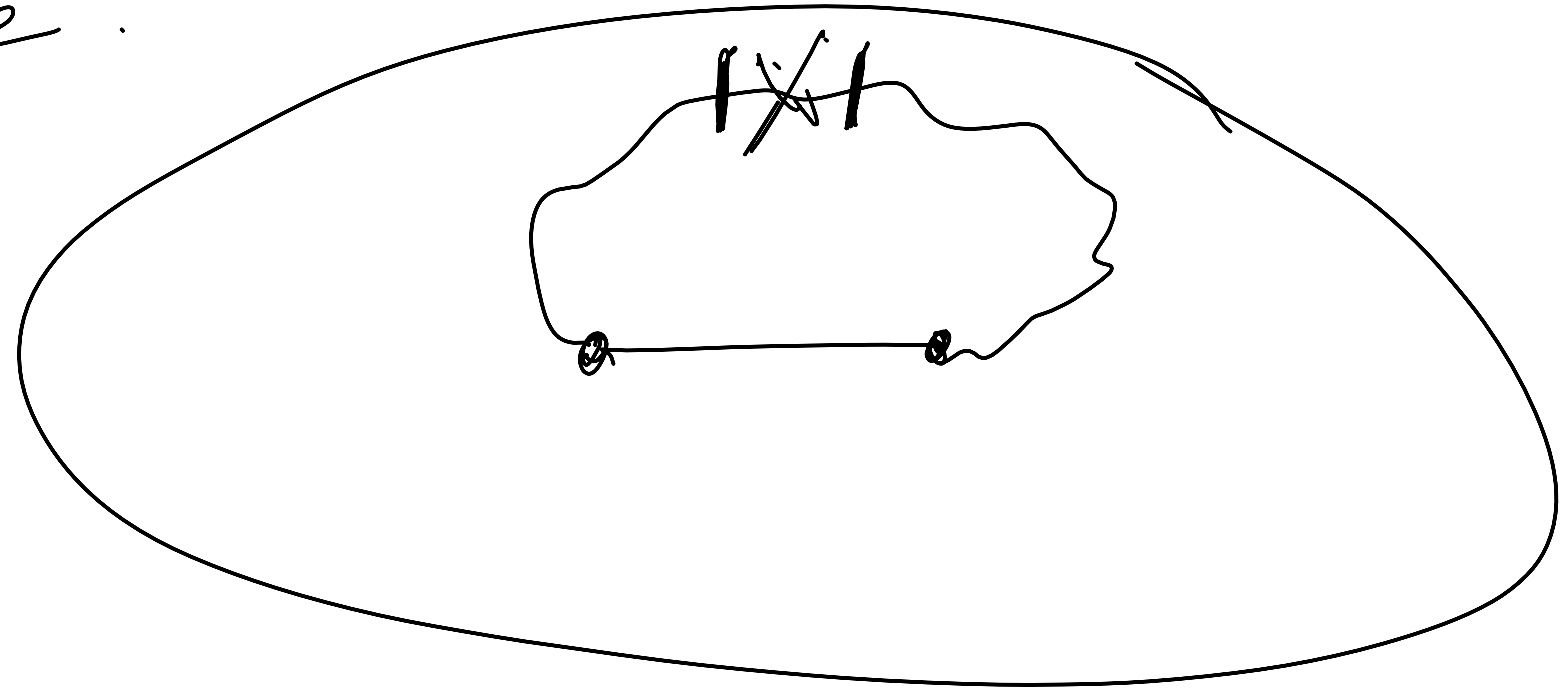
6

Claim: If $\exists$ a graph on n nodes:

- $(n-1)$ edges
- connected.

Such a graph must be acyclic & hence a tree.

Proof:



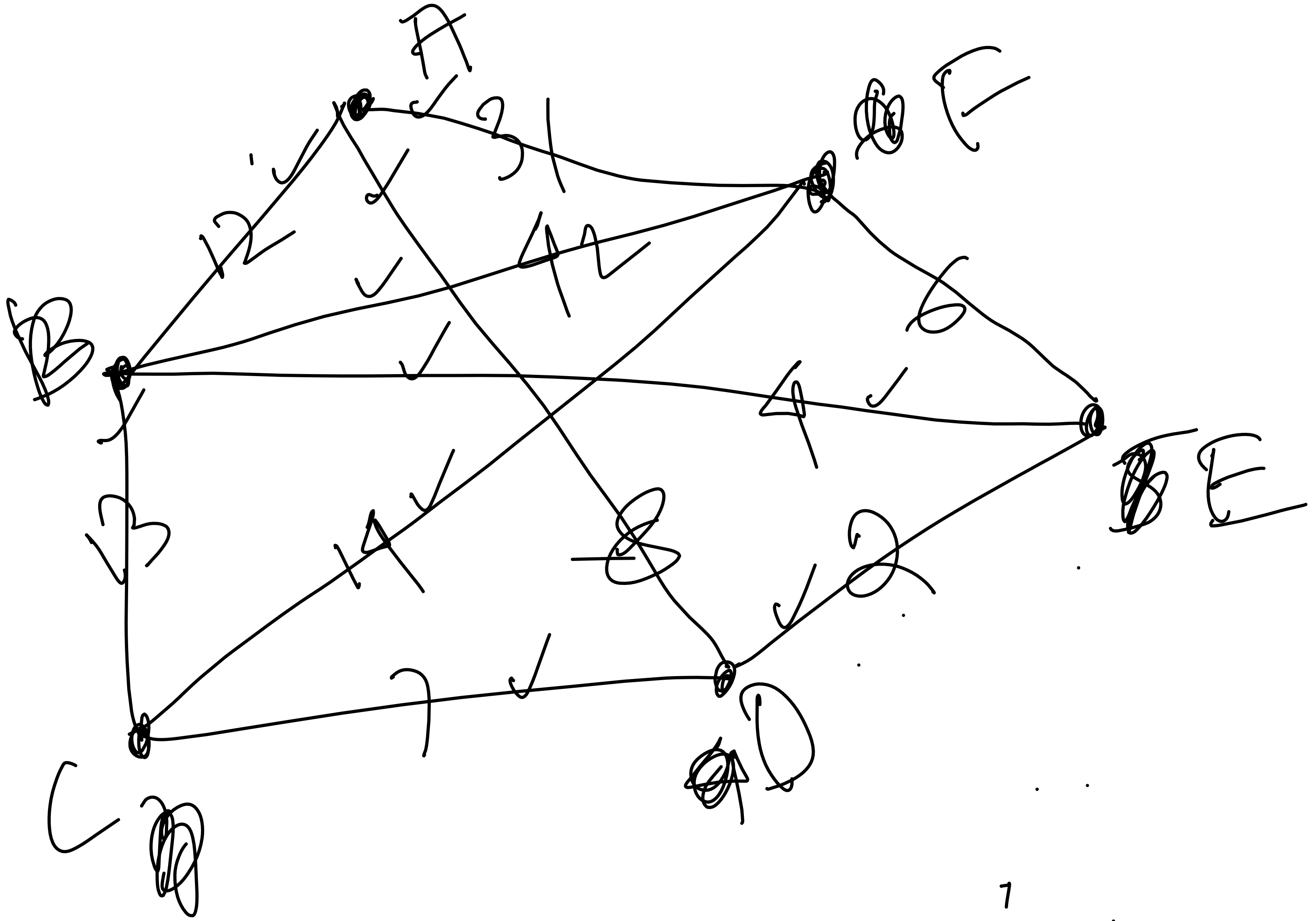
G.

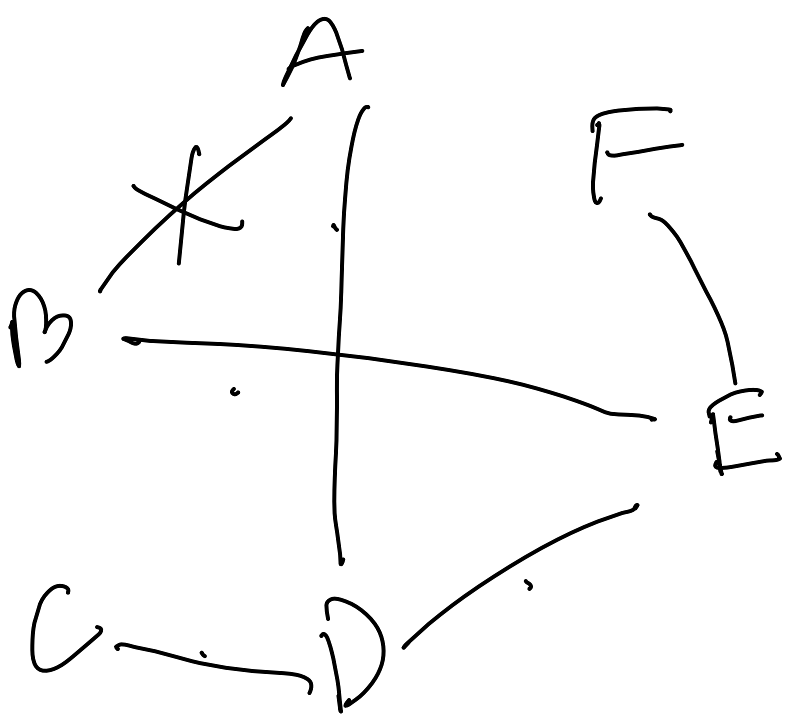
For contradiction $\exists$ cycle.

$\boxed{(n-2)}$

connected.

Contradiction?

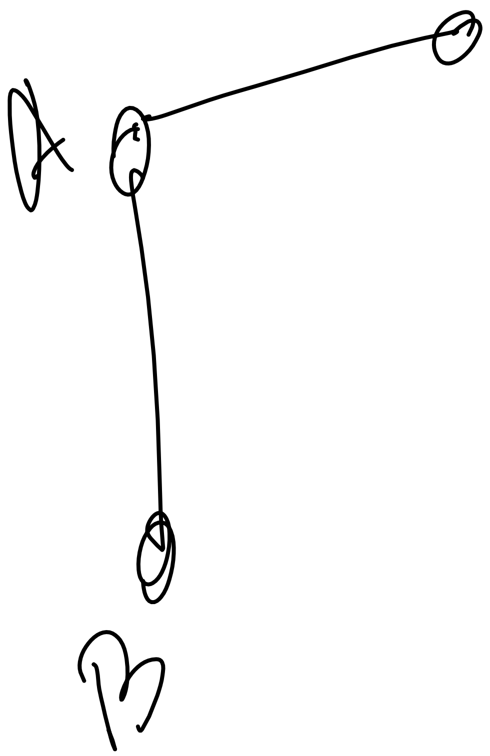


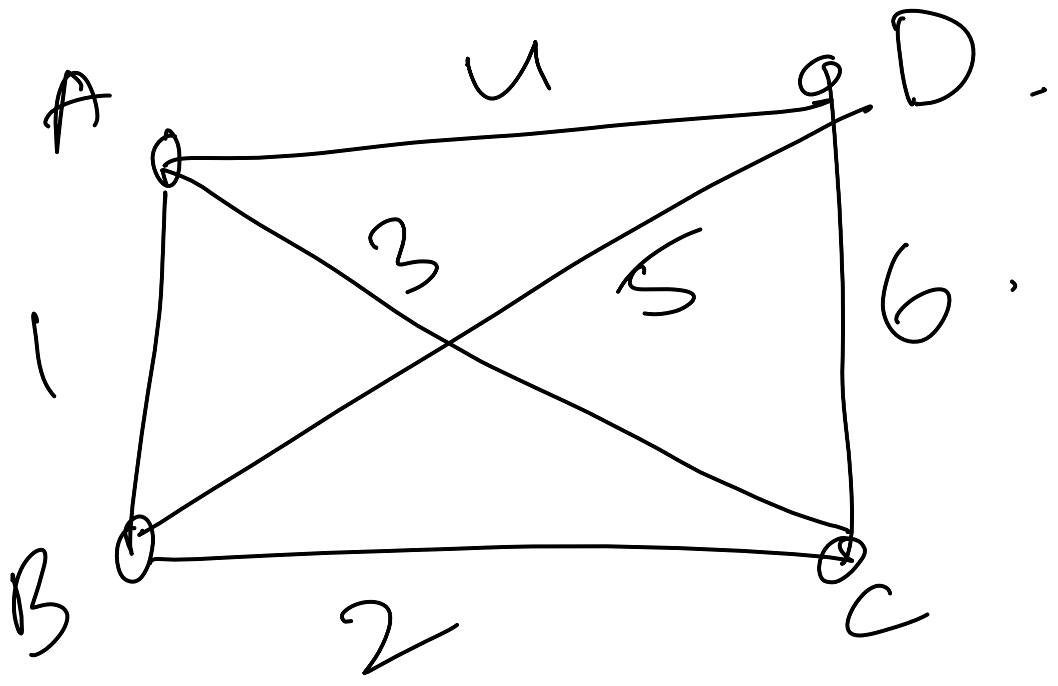


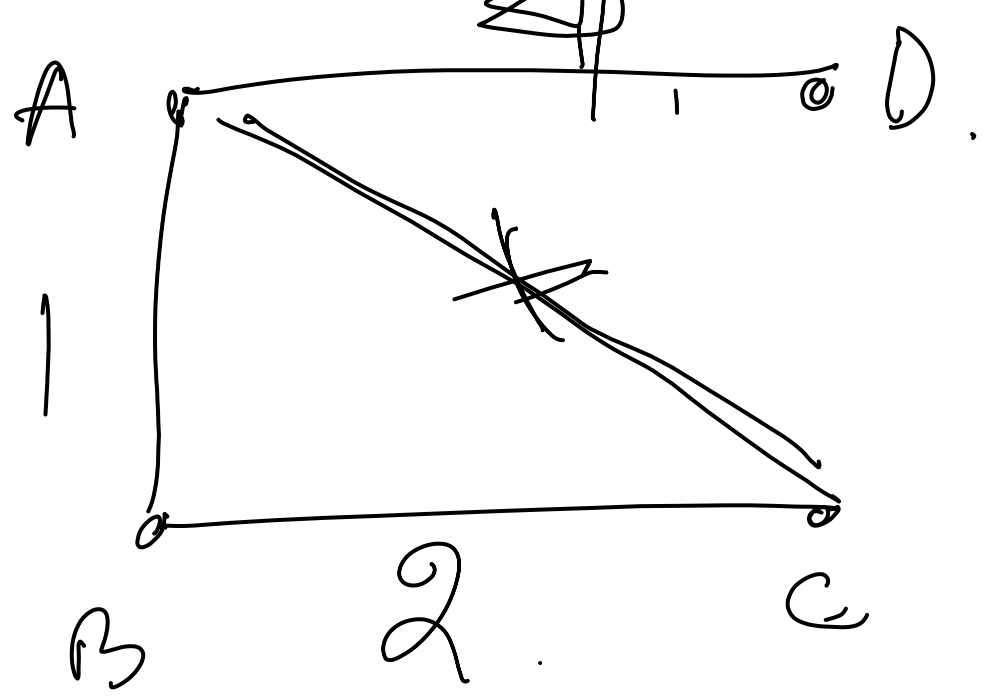
Minimum (Spanning) tree (5)

DE, BE, FE, CD, AD, BA, BC, CF, AF, BF.

2 4 6 7 8 12 13 14 31 42.

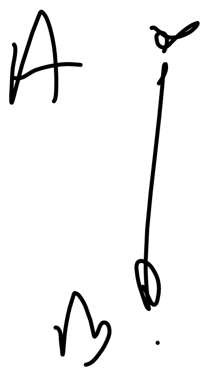






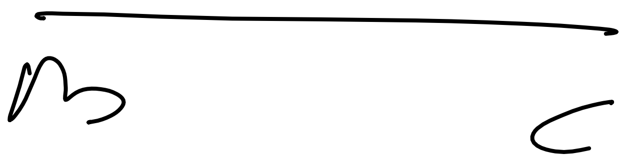
AB, BC, ~~CA~~, AD, BD, CD.

1 2 ~~3~~ 4 5 6.



Spanning $\subset$ connected $\Leftarrow$ $(n-1)$ edges

acyclic



- algorithm is simple $\longrightarrow$ locally best $\Rightarrow$ globally best.
- analysis is complicated.

Greedy algorithm

Procedure. MST (G).

// G = (V, E)

// G is given as as a weighted

// adjacency list

F ← set of edges ~~of~~ E sorted ~~hi~~ lower to higher. ← (indexed array)

i ← ~~0~~ 1

ctr ← 0. $\underline{T \leftarrow \phi}$.

while (ctr ≠ n−1)

~~if~~ e ← F[~~ctr~~ i + 1]

if (T ∪ {e} does not have a cycle)

T ← T ∪ {e}.

~~endif~~

~~ctr~~ ctr ++

end if.

i++;

end while.

return ~~T~~

(u,v)

~~T ← Merge(u,v).~~

Claim: ~~A graph~~ The graph T will always be a tree

because of previous proof

- (n-1) edges
- acyclic.

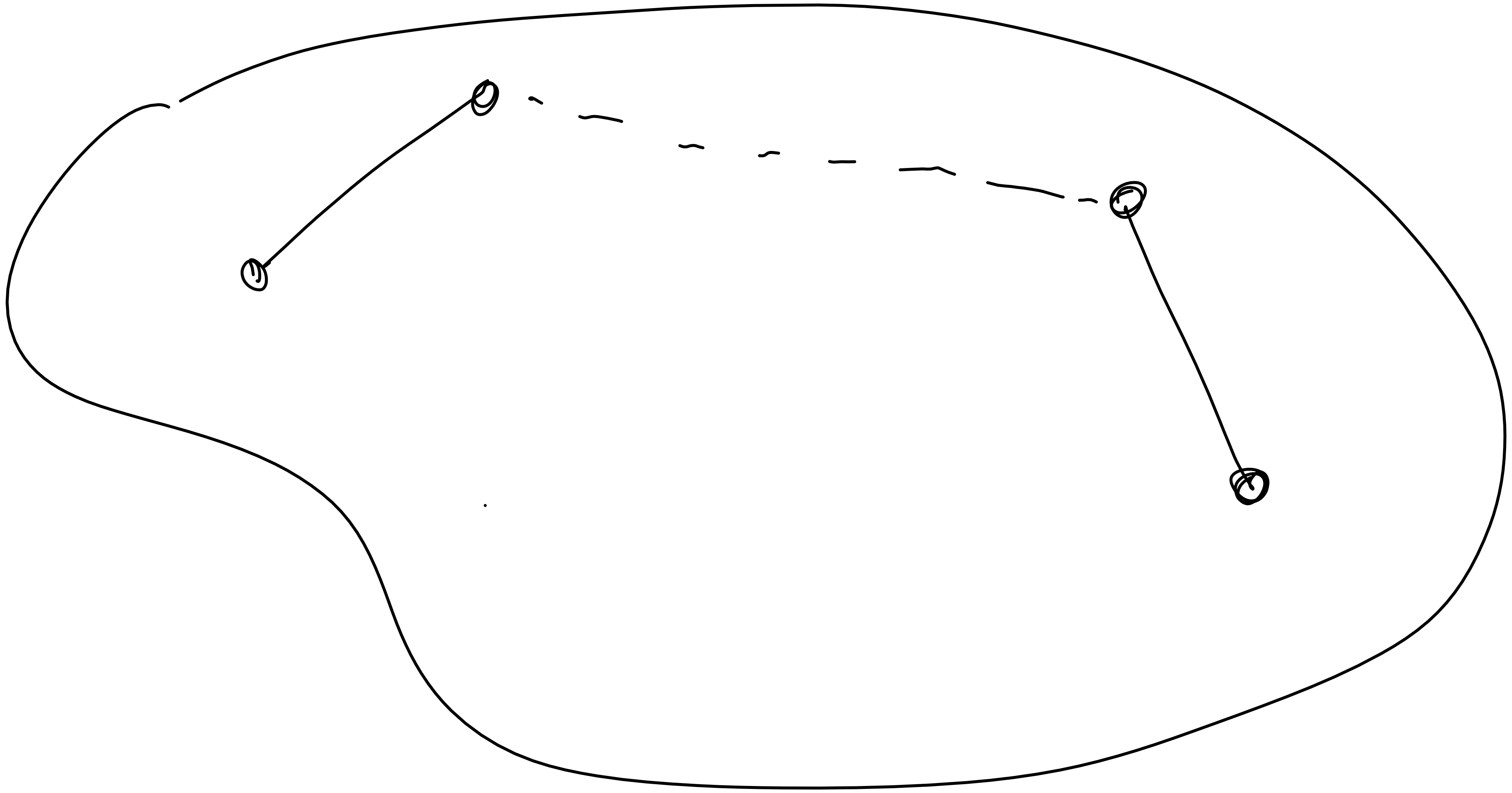
Time complexity

Sorting $\left\{ \begin{array}{l} O(|E| \log |E|) \\ = O(|E| \log |V|) \end{array} \right.$ $\quad |E| \le \binom{|V|}{2}$

Cycle checking. $\left\{ \boxed{O(\cancel{|E| + |V|}) \ \cancel{|V|}} \right.$

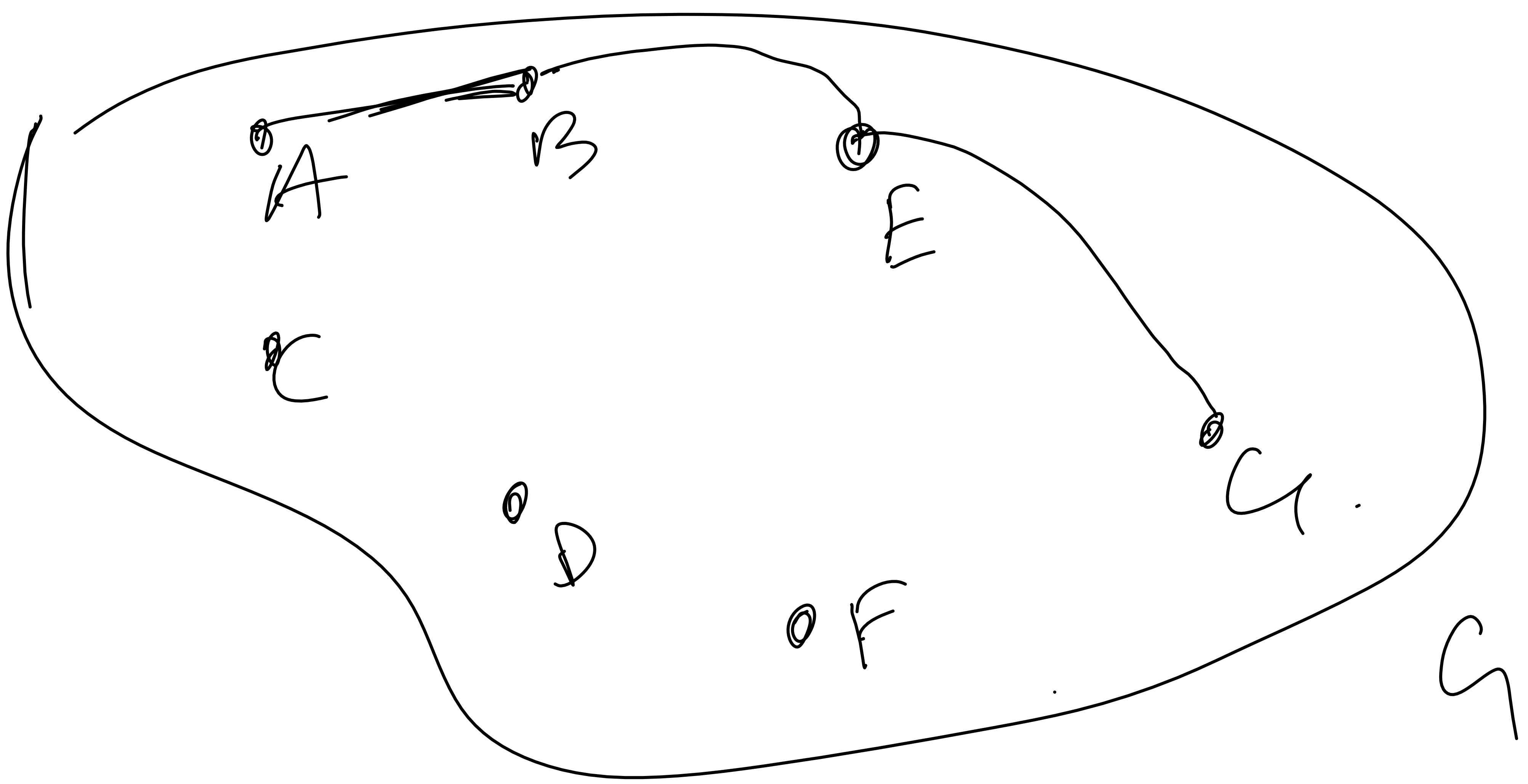
α

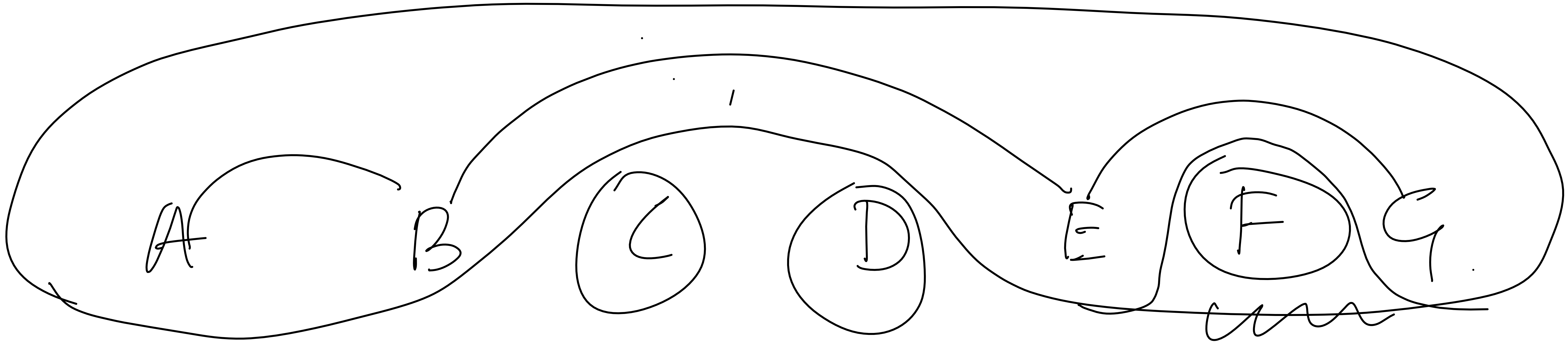
n $\quad$ n.



union-find data structure

Disjoin-set


A
B
E
C
G
D
F
G


A
B
C
D
E
F
G

Disjoint sets:

$\underbrace{S_1, \ldots, S_\ell}$. partition

- $S_i \cap S_j = \phi$
- $\bigcup_i S_i = V$

$u \in V,\ v \in V$

Q: ~~does~~ $\exists$
ℓ

Q: does (u, v) belong to same partition or not $\underbrace{\text{Find}(u)}_{S_i} \stackrel{?}{=} \underbrace{\text{Find}(v)}_{S_j}$

A: yes/no.

Q: Merge the partitions corresponding to u & v.

$\exists i \; u \in S_i$, $\exists j \; v \in S_j$

Merge(u, v) $S_i \leftarrow S_i \cup S_j$

all other partitions will remain the same.

04.09.2024

Q: