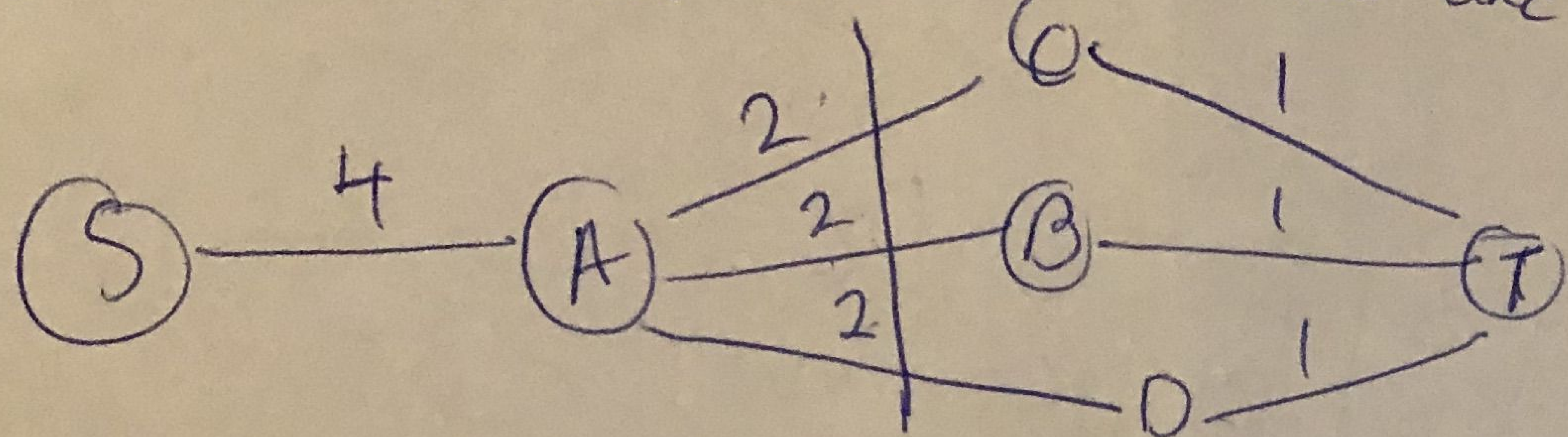
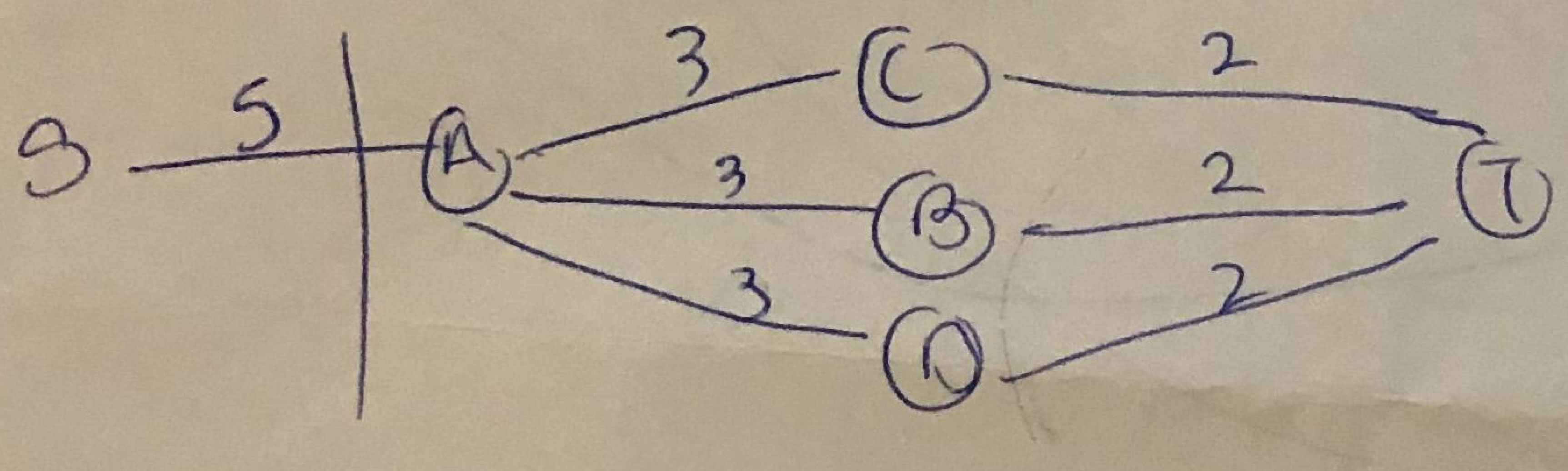
(3) Consider A Graph G where S - source and T → sink



Preposition: we increase the capacity of each edge by 1 Then (A* B*) must still be minimum s-t cut.

The minimum cut gets changed when ~~G~~ contains edges different in capacity. The minimum Cut Above has capacity 6, increasing the capacity of each edges by 1 increases the cut to 9. So the new minimum cut from S to A with a capacity of 5.



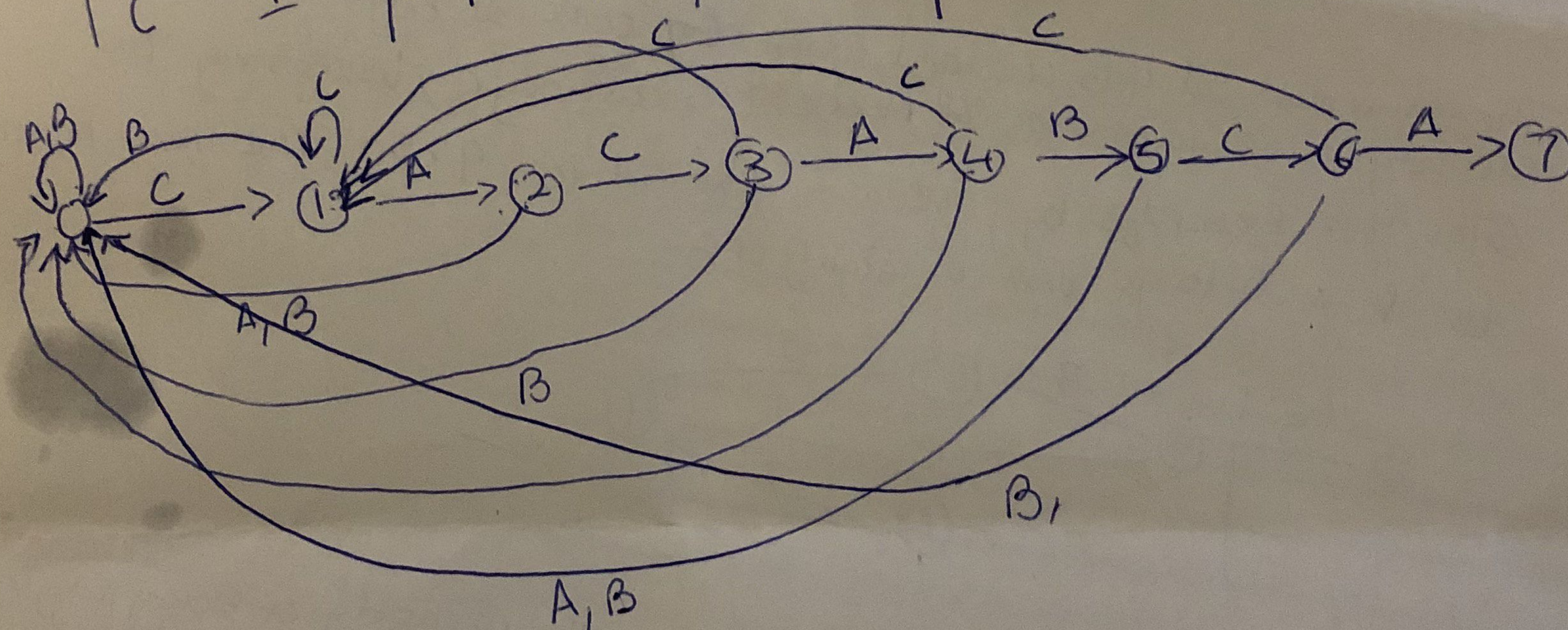
when edges have the same capacity, increasing the capacity of every edge by 1 will not change the minimum cut.

let. N_c → cut capacity
n → number of edges
C → constant
N → minimum.

If the capacity of each cut is increased by 1, we have N(C+1) and the Minimum cut is N(C+1). Assuming a cut larger than ~~N~~ N(C+1) exists, since all edges has weigh C+1, such cut must be M-Edge cut for some M>N. This would ~~Proof~~ be ~~a~~ contradicting the original graph would have capacity $M_c > N_c$. So No such M-edge cut can exist because it contradicts the assuption that N edge cut is optimal.

(4) CACABCA

		0	1	2	3	4	5	6
j		0	1	2	3	4	5	6
Pat.charAt(j)		C	A	C	A	B	C	A
dfa[ ][j]	A	0	2	0	4	0	0	7
	B	0	0	0	0	5	0	0
	C	1	1	3	1	1	6	1



(5)

① ~~for Every~~ A ship full of k-comedians
2k-ordinary People

* Each comedian has a boat
w * Each bot can carry two additional Passengers

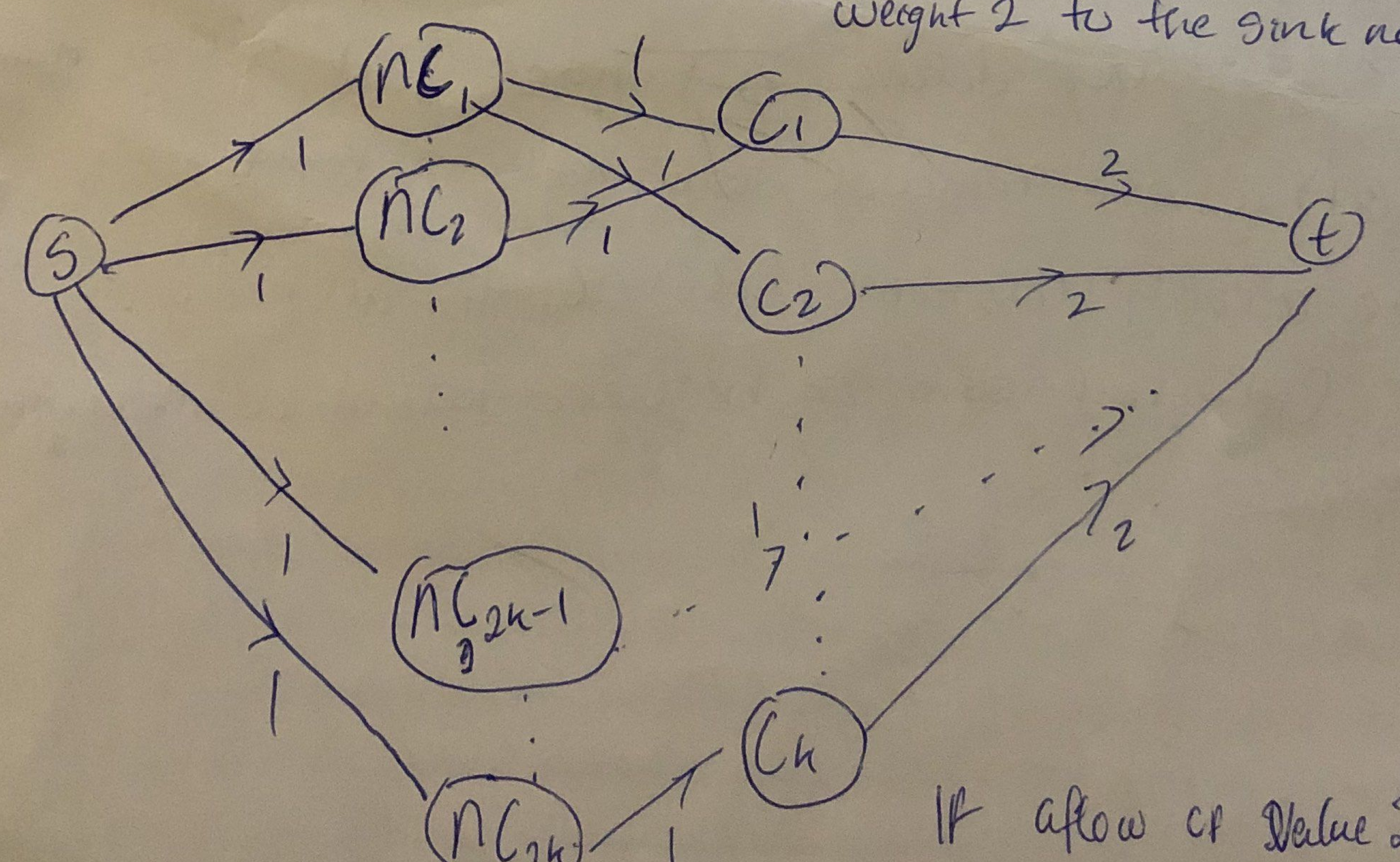
* certain non-comedians dislike certain comedians

~~$604 - Sydney~~

Let a non-comedian be represented by nc
Let a comedian ~~be~~ represented by C

~~since~~ since certain non-comedians dislike certain comedians, we draw edges from a non-comedian node to all comedian nodes whose jokes they can tolerate and assign capacity 1.

* Assign a source node s, and sink node t.
* draw edges of capacity 1 to Each non-comedian node
* for Each comedian node draw edges of weight 2 to the sink node.



If a flow of value 2k from source s Exists to sink node then it is Possible to evacuate everyone.

(2) ford-fulkerson algorithm has two main steps, first ~~is~~ labelling Process that searches for a flow augmenting Path

- from s source to t sink for which $f < c$ along all forward arcs and $f > 0$ along all backward acs

we start by ~~constructing a~~ graph A where for every vertex v in G we would have ~~vertices~~ V_i to V_o

* There is an edge from v_i to v_o of capacity c_v

* The max-flow mincut theorem for node capacity is the maximum capacity of an s-t node capacity flow of graph G is equals to the minimum capacity of an s-t node cut in G

* ~~we~~ for each edge of G (V_i, V_o) with capacity C_v, any flow in h corresponds to flow in G and all node capacities are satisfied. We define s-t node cut set of nodes $V' \subseteq V-\{s,t\}$ whose removal disconnects s from t.

The node capacity constraints $f_{in}(v) \leq C_v$ for all nodes $\therefore$ therefore ford-fulkerson can be used to find a maximum s-t flow.

5 Using A wildcard token, Given a Pattern of length m,
we hash it, we first start by hashing the first n
characters of the, if they are not the same then it is
impossible for the two strings to be the same, If
they are the same then we ~~read~~ use the normal
string for comparison or to check if they are hashed
to the same value. If not we continue and
shift over a character in the text string.