3 Prove that every even graph decomposes into cycles

Proof An Euler circuit is a circuit that uses every edge of a graph exactly once. If G is a graph in which every vertex has even degree then there exists cycles $C_1, \ldots, C_m$ of G.

If we choose a list of cycles $C_1 \ldots C_m$ which are pairwise disjoint, suppose that some component H of G has at least one edge. Since every vertex of G has even degree, it follows that every vertex in H has even degree. Since H is connected with $E(H) \neq 0$ it follow that every vertex of H has degree ≥ 2, thus every edge occurs once exactly once in $C_1 \ldots C_m$. Therefore each cycle of the cycle decomposition contributes two to the degree of each vertex in the cycle.

The degree of each vertex v in G is the sum of the degree of v of all subgraph H so it must be even.

4a) The chromatic number of G $\Rightarrow 2$.

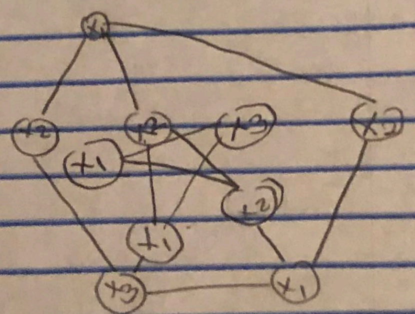
If G is bipartite, assign 1 to each vertex in one independent set and 2 to each vertex in the other set, constituting 2 colours.

4b) If the graph G is not bipartite then no given graph is not 2-colourable.

$x_1 \to$ color 1

$x_2 \to$ color 2

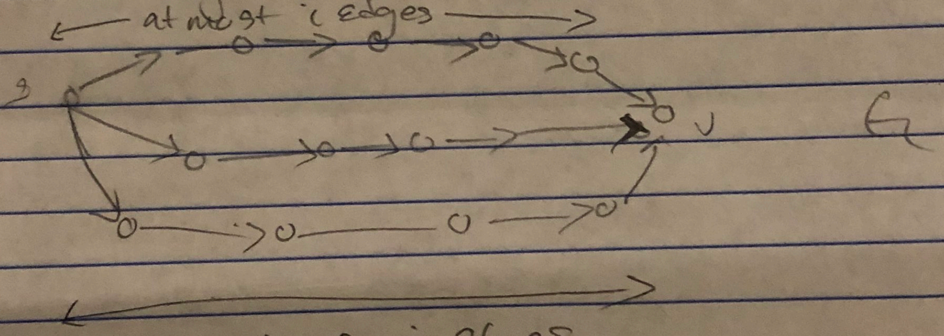
$x_3 \to$ color 3



The minimum chromatic number = 3

1) Bellman-Ford is based on the principle of relaxation where an approximation to the correct distance is gradually replaced by more accurate value until eventually reaching the optimum solution. The algorithm finds the shortest possible path of at most length i edges. Since the longest possible path without a cycle can be |V|-1 edges, a final scan is performed to all edges and if any distance is updated, the path of length |V| edges has been found which can only occur if at least one negative cycle exists in the graph.

* let l(v) be the length of the shortest path from s to v using at most i edges in the ith iteration



* A negative circle visible from s is a negative circle on a path from s to some other node v in the graph G

If the l-value of at least one node changes in round n of Bellman-Ford algorithm then there exist a negative circle visible from s. This is because the contrapositive is true: If there are no negative cycle visible from s, then the l-value does not change in round n.

1) Apply Bellman-Ford Algorithm to the graph G
2) for all edges in G:
   If l(u) + w(u,v) < l(v) then
   There is a negative cycle.

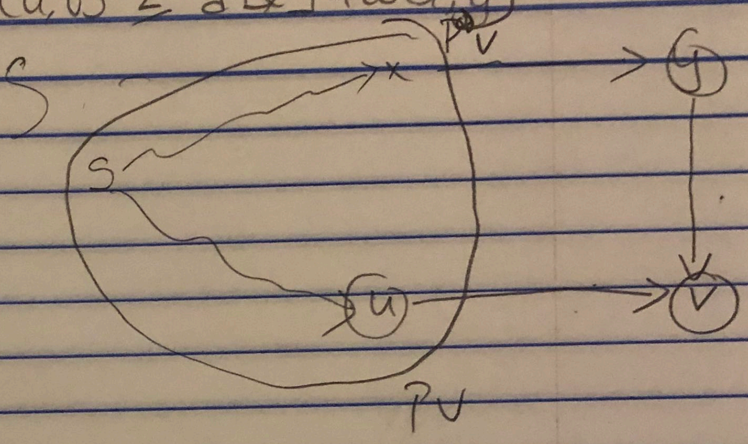
Question 2

From the lecture we saw dijkstra's proof of correctness to be,

* suppose Dijkstra adds vertex v using edge (u, v)

$u \in S$ $\quad d_u + w(u,v) \le d[x] + w(x,y)$

$v \in V/S$



$w(P'v) \geq \delta(s,x) + w(x,y)$

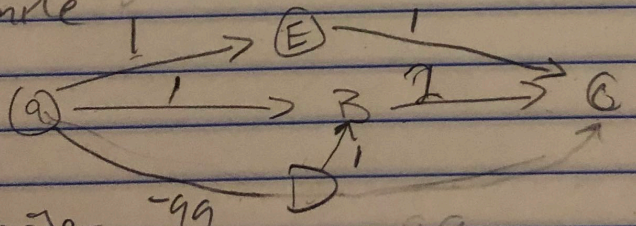
$I.H = d[x] + w(x,y)$

$= d[u] + w(u,v)$

$$I.H = \underbrace{\delta(s,u) + w(u,v)}_{w(P,v)}$$

The proof fails as, it does not add a constant amount to every Edge and thus longer path is penalized.

For example



consider

In the graph, the shortest path from a to c is a→D→B→C notice how, d[a]=0, d[B]=1, d[c]=0, d[D]=-99; d[B] then changes to -98, although d[c] is still =0 even though the shortest path to C has length -96 therefore the algorithm fails to accurately compute the distance.