Ans-1) Let G be a graph or directed graph and let s, t ∈ V (G). Then the maximum cardinality of a vertex-disjoint (resp., edge-disjoint) family of s, t-paths equals the minimum cardinality of an s, t-vertex cut (resp., edge cut). (In the former case, we assume s, t are not adjacent.)

**Proof**. First of all, an undirected graph can be considered as a digraph by replacing each edge xy with a pair of anti parallel edges xy, yx. So we may as well consider only the directed setting.

If we regard G as a network with source s and sink t, in which every edge has capacity 1, then the edge version of Menger’s Theorem is immediate from the Max-Flow/Min-Cut Theorem .

For the vertex version, we need to do a little surgery on G before applying Max-Flow/Min-Cut. The trick is to separate each vertex x ∈ V (G) \ {s, t} into an “inbox” x- and an “out-terminal” x+ with a bottleneck between them, so that only one path can pass through each vertex.

Speciﬁcally, deﬁne a digraph N by

V (N) = {s, t} ∪ {x−, x+ | x ∈ V (G) \ {s, t},

E(N) = {sx− |sx ∈ E(G)} ∪ {x+ t |xt ∈ E(G)} ∪ {x+ y− |xy ∈ E(G)}∪ {x− x+ | x ∈ V (G)},

and regard it as a network with source s and sink t and capacity function

c(e) = (1 if e =x− x+ for some x ∈ V (G),

∞ otherwise).

Then an s, t-cut in N contains only ﬁnite-capacity edges, hence corresponds to an s, t-vertex cut in G. Now applying Max-Flow/Min-Cut gives the desired result.

Ans-2) **Proof :** Using induction on the size of the cut, it sufﬁces to prove that if[S,S’] is not a bond, then[S,S’] is a disjoint union of smaller edge cuts. First suppose G is disconnected, with components G1,...,Gk. If [S,S’] cuts more than one component, we express [S,S’] as a union of edge cuts that cut only one component: let the ith cut be[S∩V(Gi),S∩V(Gi)]. This cut consists of the edges of[S,S’] in Gi, because the vertices of the other components are all on one side of the cut. Hence we may assume that [S,S’] cuts only one component of G or (equivalently) that G is connected.

An edge cut of a connected graph is a bond if and only if the subgraphs induced by the sets of the vertex partition are connected. Hence if [S,S’] is not a bond, we may assume that G[S] is not connected. Let {Gi} be the components of the induced subgraph G[S], and let Si=V(Gi). Since there are no edges between components of G[S], [S,S’] is the disjoint union of the edge cuts [Si,Si’].Since G is connected, each of these is non-empty, so we have expressed [S,S’] as a disjoint union of smaller edge cuts.

Ans-3) **Proof:** 1⇔2. A connected graph is 2-edge-connected if and only if it has no cut-edges. Cut-edges are precisely the edges belonging to no cycles.

1⇒4. By Menger’s Theorem, a 2-edge-connected graph G has two edge-disjoint x,y paths, where x,y ∈ V(G). Following one path and returning on the other yields a closed trail containing x and y.

4⇒2. Let xy be an edge. D yields a closed trail containing x and y. This breaks into two trails with end points x and y. Atleast one of them, T, does not contain the edge xy. Since T is an x,y walk, it contains an x,y path. Since T does not contain xy, this path completes a cycle with xy.

2⇒3. Choose e,f ∈ E(G); we want a closed trail through e and f. Sub divide e and f to obtain an e w graph G0, with x,y being the new vertices. Sub dividing an edge does not destroy paths or cycles, although it may lengthen them. Thus G0 is connected and has every edge on a cycle, because G has these properties. Because we have already proved the equivalence of B and D, we know that G0 has a closed trail containing x and y. Replacing the edges incident to x and y on this trail with e and f yields a closed trail in G containing e and f.

3⇒4. Given a pair of vertices, choose edges incident to them. A closed trail containing these edges is a closed trail containing the original vertices.

Ans-4) **Assumptions:** S=V(C)∩V(D), l(H)=length of a cycle or path H

**Proof By Contradiction:** If|S|<k ,it sufﬁces to construct two other cycles C’,D’ such that l(C’)+l(D’)>l(C)+l(D), because then C and D are not longest cycles in G.

For k=2,

Let e be an edge of C, and e’ an edge of D, chosen to share the vertex of S, if |S|=1. Since G is 2-connected, there is a cycle R containing both e and e’. The two portions of R between e and e’ contain paths P,Q that travel from V(C) to V(D) with no vertices of V(C)∪V(D) along the way. (If|S|=1,then one of these paths is a single vertex and has length0.) Note that since R is a cycle, P and Q are disjoint. The vertices where P and Q intersect C and D partition C and D into paths C1,C2 and D1,D2, respectively. Let C’= C1 ∪ P ∪ D1 ∪ Q and D’= C2 ∪ P ∪ D2 ∪ Q; we have

l(C’)+l(D’)= l(C)+l(D)+2l(P)+2l(Q)>l(C)+l(D).

Ans-5) Let G be a bipartite graph with bipartition X,Y. Construct a network N by adding a source s and sink t, with edges of capacity 1 from s to each x∈X and from each y∈Y to t. Orient each edge of G from X to Y in N, within ﬁnite capacity. By the integrality theorem, there is a maximum ﬂow f with integer value at each edge. The edges of capacity one then force the edges between X and Y receiving non zero ﬂow in f to be a matching. Furthermore, val(f) is the number of these edges, since the conservation constraints require the ﬂow along each such edge to extend by edges of capacity 1 from s and to t. We have constructed a matching of size val(f), so α’(G)≥val(f). A minimum cut must have ﬁnite capacity, since[s,V(N)−s] is a cut of ﬁnite capacity. Let [S,T] be a minimum cut in N, and let X’=S∩X and Y’=T∩Y. A cut of ﬁnite capacity has no edge of inﬁnite capacity from S to T . Hence G has no edge from S∩X to T∩Y. This means that (X−S)∪(Y− T ) is a set of vertices in G covering every edge of G. Furthermore, the cut [S,T] consist of the edges from s to X∩T=X−S and from Y∩S=Y−T to t. The capacity of the cut is the number of these edges, which equals |(X−S)∪(Y−T)|.We have constructed a vertex cover of size∩(S,T), so β(G)≤∩(S,T). By the Max ﬂow-Mincut Theorem, we now have β(G)≤∩(S,T)= val (f)≤α’(G). But α’(G)≤β(G) in every graph, so equality holds throughout, and we have α’(G)=β(G) for every bipartite graph G.