HOMEWORK 6

SOLUTIONS TO 2.3 PROBLEMS

2.3.6 Assign integer weights to the edges of K_n. Let the weight of a cycle be the sum of the weights of its edges. Prove that all cycles have even weight if and only if the subgraph formed by the edges with odd weight is a spanning biclique. (Hint: Show that every component of the subgraph consisting of the edges with even weight is a complete graph.)

Proof. Consider the the subgraph H consisting of the edges with even weight. If one component H of this subgraph were not complete, then it must have two non-adjacent vertices u and v. Then the cycle in K_n which traces a path in H from u to v and then the edge in K_n would have odd weight. So if every cycle in K_n is even, the components of the subgraph consisting of the edges with even weight are cliques. There must be at least one even edge, because every K_n for $n \geq 3$ has 3-cycles, and there can be at most two components in H because otherwise the cycle which traces a path in H from u to v, followed by an edge to a second component, and then to a third, and back to u would be odd. The complement of H is the subgraph of K_n formed by the edges with odd weight, so the vertices in those components form independent sets, and every edge connecting vertices from one set to the other is in $\bar{H}$. So $\bar{H}$ is a biclique.

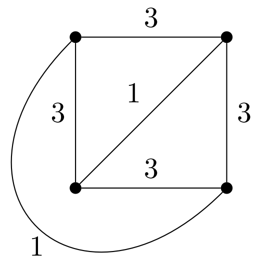
Now, if the if the subgraph formed by the edges with odd weight is a spanning biclique, then every cycle in K_n must have an even number of odd weighted edges, so every cycle is of even weight. □

2.3.20 *Minimum diameter spanning trees.* **A MDST is a spanning tree where the maximum length of a path is as small as possible. Intuition suggests that running Dijkstra's Algorithm from a vertex in the center will produce an MDST, but this may fail.**

(a) **Construct a 5-vertex example of an unweighted graph such that Dijkstra's Algorithm can be run from some vertex in the center and produce a spanning tree that does not have minimum diameter.**

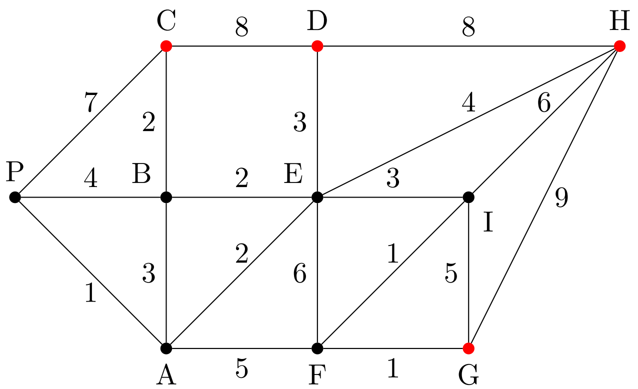
Since ties are broken arbitrarily, Dijkstra's algorithm can produce a P_5 from the 5-cycle with an extra edge if run from a vertex not part of the three-cycle (the whole graph is the center). (Prove by executing the algorithm, noting that a MDST has diameter 3, not 4.)

(b) **Construct a 4-vertex example of a weighted graph such that Dijkstra's Algorithm cannot produce a MDST when run from any vertex.**

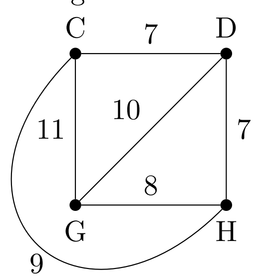


(Prove by executing the algorithm, noting that a MDST has diameter 5, not 6.)

2.3.23 Every morning the Lazy Postman takes the bus to the Post Office. From there, he chooses a rout to reach home as quickly as possible. Below is a map of the streets along which he mush deliver mail, giving the number of minutes required to walk each block whether delivering mail or not. P **denotes the post office and** H **denotes home. What must the edges traveled more that once satisfy? How many times will each edge be transversed in the optimal route?**



Solution. The red vertices (C, D, G, and H) are the ones of the 'wrong' parity (P is odd, but it gets visited an odd number of times; H is even, but is gets visited an odd number of times.) So we want to find minimal pairings in the complete graph on C, D, G, and H with edges labeled by minimal paths connecting them:



The minimal is CD and GH, so we're looking to duplicate edges along paths of length 7 from C to D and of length 8 from G to H. This can be dome by retracing CD, BE, ED, and GE, FI, IH. All other edges will be traced exactly once.

2.3.24 Explain why the optimal trails pairing up odd vertices in an optimal solution to the Chinese Postman Problem may be assumed to be paths. Construct a weighted graph with four odd vertices where the optimal solution to the Chinese Postman Problem requires duplicating the edges on two paths that have a common vertex.

Solution: Every u, v trail contains a u, v path, so if the trail is not a path, then we over duplicate edges.