Robert Christie

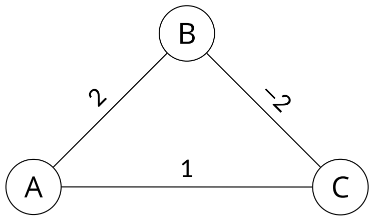
MATHS 326

S1 2024

Assignment 2
Due: 19-04-2024

Q1

Consider the following weighted graph:



Applying Dijkstra's Algorithm starting from $v = A$:

- Initial stage: We set:

$$R = \varnothing, \quad \delta(v) = \begin{cases} 0 & v = A \\ \infty & v = B \\ \infty & v = C \end{cases}$$

  Also set A to be the current vertex.

- Iteration 1: Add A to R, and update δ giving:

$$R = \{A\}, \quad \delta(v) = \begin{cases} 0 & v = A \\ 2 & v = B \\ 1 & v = C \end{cases}$$

  Then we set C as the current vertex as it has the smallest δ of the unvisited vertices.

- Iteration 2: We add C to R, and update δ giving:

$$R = \{A, C\}, \quad \delta(v) = \begin{cases} 0 & v = A \\ -1 & v = B \\ 1 & v = C \end{cases}$$

  Now that C has been visited, the algorithm will not alter $\delta(C)$ any further.

Thus, the final value of $\delta(C) = 1$. However, the actual shortest path from A to C is A—B—C with length 0. So Dijkstra's algorithm doesn't always find shortest paths if positive and negative edge weights are allowed.

Q2

Let $G = (V, E, w)$ be a weighted graph with weight function w.

Proof by contradiction:

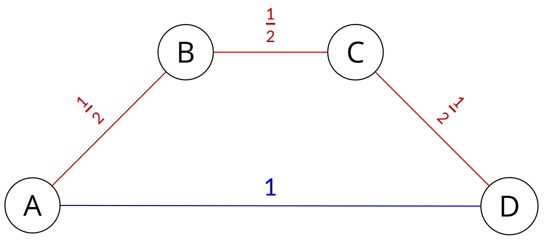
Assume that G does not have a unique MST, thus we can find two distinct spanning trees T and T' with the same minimal weight. Since T and T' are distinct, there must exist an edge in one (but not both) of T and T', let e be such an edge with minimal $w(e)$ and WLOG assume e is in T but not in T'.

Let $e = \{u, v\}$, since T' is also spanning, there is some other path from u to v in T', so adding e to T' would create a cycle. Because T is a tree, it cannot contain the cycle, so some other edge e' of the cycle is not in T.

Since e' is also in one but not both of T and T' and e was the smallest edge with this property, we must have $w(e') > w(e)$. However, if we remove e' from T' and insert e, we have constructed a new MST with smaller weight, meaning T and T' were not minimal. This is a contradiction so G must have a unique MST.

Q3

(a) Counter example: Consider the following graph:



In the original weighting, the shortest path from A to D is given by the blue path with weight 1. Replacing each edge weight with the square would make the red path shortest with weight $\frac{3}{4}$, the blue path still has length 1.

(b) Let $G = (V, E, w)$ be a weighed graph with weight function $W : E \to \mathbb{R}^+$.

Consider any T_0 of G and apply Kruskal's algorithm, if edges have the same weight, choose edges in T_0 first before looking at other edges. This still follows the definition of Kurskal's algorithm as the order of edges with the same weight is undecided.

We claim that this modification of Kruskal's algorithm will output a T with $T = T_0$ when T_0 is an MST.

Proof by contradiction:

Assume that T_0 was not output, therefore at some point an edge e was added to T_0 where e not in T_0. Consider the first such edge added.

Until e was added, T is a subgraph of T_0, and all edges in T_0 of weight at most $w(e)$ are already in T. Since Kruskal's algorithm added e, this did not create a cycle in T.

Now consider adding $e = \{u, v\}$ to T_0, this must create a cycle since T_0 is connected, so there is already a path from u to v.

When e was added, T already contained all the edges of C with weight at most $w(e)$ and did not contain a cycle. Hence, we can break the cycle by removing an edge e' from C with $w(e') > w(e)$.

This creates a spanning tree with smaller weight than T_0, we have a contradiction as T_0 was already an MST.

Therefore, Kruskal's algorithm will output T with $T = T_0$ when T_0 is a MST.

Now consider some strictly increasing function $f : \mathbb{R}^+ \to \mathbb{R}^+$. Consider any MST T_0 in G and $G' = (V, E, f \circ w)$.

Consider Kurskal's algorithm applied to G and G' choosing edges in T_0 first when edges have the same weight.

Step 1: Both initialise T to the same spanning subgraph with no edges.

Step 2: Both algorithms will consider the edges in the same order as f is increasing, so it preserves the order of w.

Step 3: We can apply induction (base case step 1) to see that the same edges will be added as at each iteration: If the same edges were added to T previously, then e creates a cycle in T as a subgraph of G' if and only if it created a cycle in G as cycles aren't affected by the weighting function.

Therefore, Kurskal's algorithm on G' will output the same $T = T_0$ as it does on G.

Since x^2 and $\sqrt{x}$ are strictly increasing functions $\mathbb{R}^+ \to \mathbb{R}^+$, applying our previous working:

- Any MST in G with weighting function w is also a weighting function in G' with weighting function $w' = w^2$.
- We also see that any MST in G', is a MST in G with weighting function $w = \sqrt{w'}$.

Thus, we have shown equivalence.

Q4

First we show "$1 \implies 2$":

Assume a tree $T = (V, E)$ has a perfect matching. For any $v \in V$, there is some $u \in V$ matched to v. This splits T into $n \geq 1$ connected components.

One of these components C_1 must contain u, there must be no other edge from C_1 to v in T otherwise T would contain a cycle.

Since u was matched to $v \notin C_1$, any edge from C_1 to u was unmatched so restricting M onto $C_1 - u$ does not break any matching giving $C_1 - u$ a perfect matching.

Therefore, C_1 has an odd number of vertices.

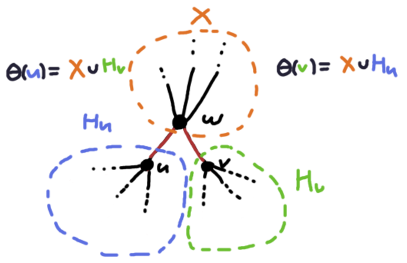
Each remaining connected C_i did contain a vertex matched to v. Since no matched pair has a single vertex in C_i, restricting M to C_i gives a perfect matching so C_i has an even number of vertices.

Thus, $o(T - v) = 1$.

Now to show the other direction "$2 \implies 1$", we assume $o(T - v) = 1$. Observe and define the following:

- $T - v$ must contain exactly one odd connected component, call this component $\Theta(v)$.
- As T was connected, each connected component C_i contains a $u_i \in V$ adjacent to v in T.
- There is exactly one such u_i in each component. If this wasn't the case then this would create a cycle in T as u_i and u_i' are already connected in C_i, this isn't possible as T is a tree.
- Construct the set M containing $e = \{v, u\}$ for each $v \in V$, where $u \in \Theta(v)$ is adjacent to v in T.

We claim that the edges in M are mutually disjoint. Proof by contradiction:



Assume that two edges in M are not disjoint, so some $u, v \in V$ were both paired to some w.

Let H_u be the subgraph of T induced by all the vertices of the even connected components of $T - u$ and the vertex u. Notice that H_u is connected as u connected to each of the connected components. Note that H_u has an odd number of vertices as it contains both the even connected components and a singular u (all of which are disjoint). Define H_v in the same way.

For any $x \in \{u, v\}$ then only x could be connected to a vertex outside of H_x, otherwise. If this were not the case and H_x had an edge to some other vertex y, then this edge would form a cycle from the existing path through x to y, however T cannot contain cycles.

The subgraphs H_u, H_v are disjoint since $\Theta(u)$ contains w which is still connected to v in H_v, thus H_v is a subgraph of $\Theta(u)$ which is not contained in H_u.

Notice that both H_u and H_v contain an odd number of vertices as they contain the even connected components in $T - u$ or $T - v$ as well as u or v respectfully.

Now consider $T - w$, both H_u and H_v are connected components in $T - w$ as they are not contained in any larger connected component and the only edge in T leaving either subgraph contained w.

Both H_u and H_v have an odd numbers of vertices so $o(T - w) \geq 2$ which is a contradiction.

Therefore, M is a matching by definition, however M is also a perfect matching as it clearly contains every $v \in V$.

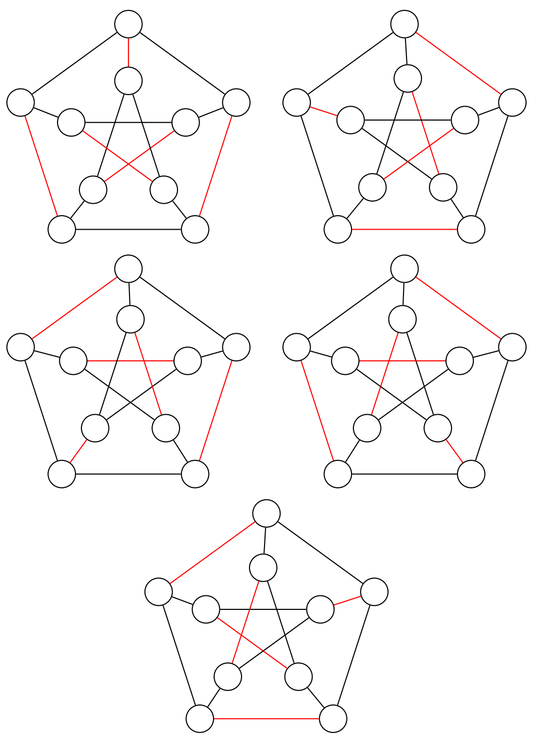
Q5

(a) Since $G = (V, E)$ is regular, G is d-regular for $d = \Delta(G)$. Since G is class 1, there is a d-edge colouring $\lambda : E \to C$ with $|C| = d$. For any edge e_0 of G, consider the matching give by:

$$M = \left\{e \in E \,:\, \lambda(e) = \lambda(e_0)\right\}$$

By definition of edge-colouring, the edges of M are mutually disjoint and therefore a valid matching. Since each degree has degree d, there are d edges with each of the colours including $\lambda(e_0)$. Thus, every vertex must be contained in M making M a perfect matching. Thus, any e_0 is matchable.

(b) The Petersen graph is 3-regular and was shown to have chromatic index 4 in *Assignment 1*, thus it is regular and class 2. Consider the following matchings on the Petersen graph where the red edges are the matched edges:



Since every edge is contained in at least one of the matchings above, no edge is unmatchable.

Q6

We use the following results from the course book:

- *Exercise 1.26*:

$$\alpha(G) + c(G) = n = |V|$$

- *Theorem 2.15 (Konig's Theorem)* if G is a bipartite graph, then $m(G) = c(G)$ where $m(G)$ is the size of a maximum matching.

Proving the inequality:

By definition, no independent set $I \subseteq V$ is larger than a maximum independent vertex set, so $|I| \leq \alpha(G)$.

Similarly, no matching $M \subseteq E$ is larger than a maximum matching so $|M| \leq m(G)$.

Thus, for any bipartite graph G:

$$\begin{aligned} |I| + |M| &\leq \alpha(G) + m(G) \\ &= \alpha(G) + c(G) && \text{By Theorem 2.15} \\ &= |V| && \text{By Exercise 1.26} \end{aligned}$$

So $|I| + |M| \leq |V|$.

Now showing that equality holds if and only if I, M are maximum.

First assume that I and M are maximum independent sets/matchings, then:

$$|I| = \alpha(G), \qquad |M| = m(G)$$

Therefore:

$$\begin{aligned} |I| + |M| &= \alpha(G) + m(G) \\ &= \alpha(G) + c(G) && \text{By Theorem 2.15} \\ &= |V| && \text{By Exercise 1.26} \end{aligned}$$

So $|I| + |M| = |V|$.

We show the other direction of equivalence using the contrapositive. Assume for some bipartite G, $|M|$ is not a maximum matching or $|I|$ is not a maximum independent set, then:

$$|I| < \alpha(G) \quad \vee \quad |M| < m(G)$$

Therefore:

$$\begin{aligned} |I| + |M| &< \alpha(G) + m(G) \\ &= \alpha(G) + c(G) && \text{By Theorem 2.15} \\ &= |V| && \text{By Exercise 1.26} \end{aligned}$$

So $|I| + |M| < |V|$, equality does not hold, so the contrapositive is also true.

We can also prove *Exercise 1.26*:

Consider a maximum independent set I on $G = (V, E)$. Thus, $\alpha(G) = |I|$. By definition, every other vertex in G is adjacent to a vertex in I (otherwise I is not a maximum). Let $C = V \setminus I$. No edge contains exclusively vertices in I, so every edge contains at least one vertex in C making C a vertex cover. Thus, $c(G) \leq |C|$ so:

$$\alpha(G) + c(G) \leq |I| + |V \setminus I| = |V|$$

Now assume that C is a minimum vertex cover on $G = (V, E)$. Thus, $c(G) = |C|$. By definition, no edge $e \in E$ has $e \subset V \setminus C$. Therefore, $I = V \setminus C$ is an independent set and $\alpha(G) \geq |I|$. Therefore:

$$\alpha(G) + c(G) \geq |I| + |V \setminus I| = |V|$$

For both inequalities to hold we must have:

$$\alpha(G) + c(G) = |V|$$

Q7

We show equivalence by proving the reverse implication and its contrapositive:

Start with "$2 \implies 1$", assume that $G = (V, E)$ has a perfect matching M. Let $P \subseteq V$ be the path created by the game.

Induction on Bob's turns.

- Base case (First turn): Since the initial v is matched, Bob can choose the edge $e = \{v, u\} \in M$. Alice must choose an unmatched edge adjacent to e (or loose), by definition of a matching.

- Induction Step: Alice either looses or chooses an unmatched edge $e = \{u, v\}$ where $u \in P$ and v was not already in P. As M is a perfect matching, there is always a single $e' = \{v, w\} \in M$ that matches v. To see that $w \notin P$:

  - If Alice had already added w to P, then on the next turn, Bob would have chosen e' already.

  - If Bob was the one that added w to P, then since Bob only chooses matched edges, v was also already in P, so Alice could not have chosen e.

  Therefore, Bob can choose $e' = \{v, w\} \in M$, let S be the set of edges Alice could have chosen from before, now she can choose from some:

$$S' \subset S \cup \{\{w, x\} : x \in N(w)\}$$

Alice must choose an unmatched edge as all edges in S were unmatched and $\{\{w, x\} : x \in N(w)\}$ must also only contain unmatched edges as there are adjacent to e' which is matched.

Since finite turns occur, Alice must eventually lose.

Now we show "$\neg 2 \implies \neg 1$", assume $G = (V, E)$ does not contain a perfect matching, then let M be a maximum matching with some unmatched vertex $v \in V$, Alice chooses v as the starting point.

Claims: The path P is always alternating and Bob can never choose a matched edge.

To prove these claims we use induction on Bob's turns:

- Base case (First turn): Bob must start by choosing an unmatched edge $e = \{v, u\}$ as v is unmatched. Path is alternating.
- Induction Step: Assume that u, the last vertex added to the path by Bob was unmatched, then the path contains an augmenting path from v to u as both u and v are unmatched and the path is alternating. This contradicts M been a maximum matching. Hence, u is matched.

  We can verify that Alice can choose the edge $e = \{u, w\} \in M$ that matches u, Bob cannot have already chosen this edge as he only chooses unmatched edges. If Alice already chose this edge then in Bob's last turn he would have added an edge to v which would have already been in P which is not a valid move.

  Since $e \in M$ is matched, by the definition of a matching, Bob is forced to choose an unmatched edge or loose if there are none. As Alice and Bob alternate adding matched and unmatched edges, the path is alternating.

Since the game can only go for finite steps, eventually Bob must lose meaning he does not have a winning strategy.