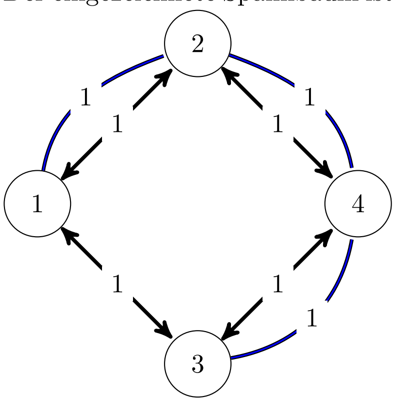
Name 1, Gruppe X
Name 2, Gruppe Y

Lösungen zum 0. Übungsblatt (AGT 21)

Aufgabe 1:

a) Diese Aussage ist korrekt. In einem zusammenhängenden Graphen besucht die Breitensuche jeden knoten genau ein mal. Der Breitensuchenbaum enthält genau $|V| - 1$ Kanten (da jeder Knoten über genau eine Kante besucht wird) und hat somit ein gesamtes Gewicht von $|V| - 1$. Da der minimale Spannbaum ebenfalls $|V| - 1$ Kanten enthalten muss (Graph ist zusammenhängend) und alle Kanten das Gewicht 1 haben, muss das Gewicht jedes minimalen Spannbaums ebenfalls gleich $|V| - 1$ sein. Da also alle möglichen Breitensuchenbäume das selbe gewicht wie der minimale Spannbaum (bzw. die minimalen Spannbäume) hat, ist jeder Breitensuchbaum somit ein minimaler Spannbaum.

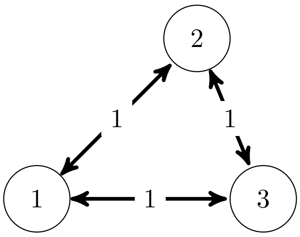
b) Der blau gezeichnete minimale Spannbaum vom Graphen G ist kein Breitensuchbaum mit Quelle 1. Dies ist gegeben da wenn man mit der Breitensuche von 1 startet, man zunächst den Knoten 2, dann den Knoten 3 besuchen würde (oder umgekehrt). Der eingezeichnete Spannbaum ist also kein Breitensuchbaum



mit Startpunkt 1.

Aufgabe 2:

a) Die Skize zeigt den kleinsten nicht zweifärbbaren Graphen.
Graph mit 1 Knoten: Eigenschaft trivial erfüllt
Graph mit 2 Knoten: Färbe Knoten 1 rot, Knoten 2 blau
$\Rightarrow$ Der kleinste nicht zweifärbbare Graph hat 3 Knoten.



b) Der Algorithmus hat eine Laufzeit von O($|V| + |E|$) da unsere erste For-Schleife jeden Knoten genau einmal behandelt (wir gehen davon aus das c für jeden Knoten eine Färbung besitzt) und wir in der zweiten Schleife ein mal über alle Kanten iterieren.

TesteFärbung(G, c)
    // Map die für jeden Knoten die Färbung speichert
    f = {}
    foreach $k \in c$ **do** Alle Knoten färben
        // Färbung des Knoten speichern
        f[k] = k.color
    foreach $e \in V$ **do** Testen ob alle Knoten passend gefärbt sind
        if *f[e.start] == f[e.end]* **then** Wenn die Knoten am Start und Ende der Kante nicht unterschiedlich gefärbt sind
            return false
    return true

c) Der Algorithmus läuft in O($|V| + |E|$) da von jedem Knoten alle ausgehenden Kanten behandelt werden.

Färbbar(G)
    c = 0
    foreach $k \in V$ **do**
        if $k.c == (c + 1\%2)$ **then** Wenn der aktuelle Knoten schon eine Färbung hat die anders ist als die aktuelle Färbung
            Return false
        // Den aktuellen Knoten in der aktuelle Farbe färben k.color = c
        foreach $n \in Adj[k]$ **do**
            if *n.color == k.color* **then** Wenn beide Knoten die selbe Farbe haben
                Return false
        // C von 0 auf 1 bzw von 1 auf 0 setzen
        c = c + 1 % 2
    return true

Aufgabe 3:

a) Die Knoten des Graphen sind alle Felder des Schachbrettes. Von einem Knoten (also Schabrett-Feld) gibt es eine Kante zu all den Feldern, auf die ein Springer sich (vom aktuellen Feld aus) bewegen kann.