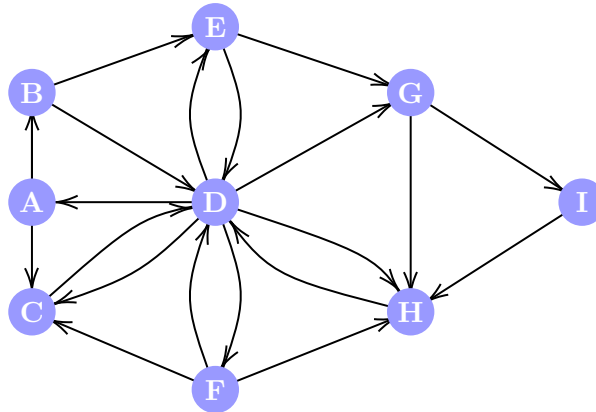


Übungsblatt 9

Abgabe via Moodle.
Deadline Fr. 7ter July

Aufgabe 1 (*Breitensuche*, 1 + 1 + 1 + 1 *Punkte*)

Gegeben sei folgender gerichteter Graph mit der Knotenmenge $\{A, B, \dots, I\}$:



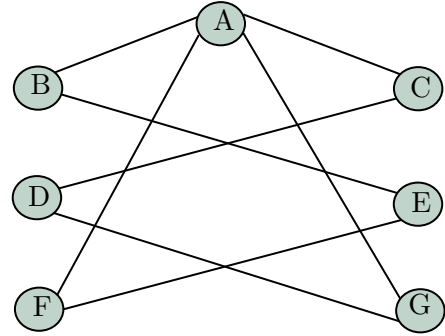
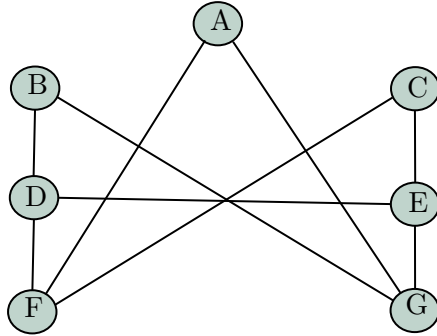
In diesem Graph werde nun eine Breitensuche durchgeführt, und zwar ausgehend vom Knoten A .

1. Zählen Sie die Knoten des Graphen in einer Reihenfolge auf, in der diese von einer Breitensuche jeweils zum ersten mal berührt werden. Heben Sie im Graph außerdem alle Kanten (z.B. farbig) hervor, die Bezüglich dieser Breitensuche *tree* Kanten sind.
2. Sind im Graph bzgl. dieser Breitensuche irgendwelche *backward* Kanten vorhanden? Wenn ja, heben Sie diese hervor.
3. Sind im Graph bzgl. dieser Breitensuche irgendwelche *cross* Kanten vorhanden? Wenn ja, heben Sie diese hervor.
4. Sind im Graph bzgl. dieser Breitensuche irgendwelche *forward* Kanten vorhanden? Wenn ja, heben Sie diese hervor.

Aufgabe 2 (Bipartite Graphen, 2 + 6 + 2 Punkte)

Ein ungerichteter zusammenhängender Graph $G = (V, E)$ mit $|V| \geq 1$ heißt **bipartit**, wenn die Knotenmenge V so in zwei Mengen V_1 und V_2 aufgeteilt werden kann, dass für jede Kante $(u, v) \in E$ gilt $u \in V_1$ und $v \in V_2$ oder $v \in V_1$ und $u \in V_2$. Mit anderen Worten existiert eine Zerlegung von V , so dass Kanten nur zwischen Knoten aus V_1 und V_2 bestehen, nicht aber zwischen Knoten der V_i selbst.

1. Zeigen oder widerlegen Sie jeweils die Bipartitheit der folgenden beiden Graphen!



2. Entwickeln Sie einen Algorithmus, der in $O(n + m)$ entscheiden, ob ein ungerichteter zusammenhängender Graph bipartit ist. Im Fall der Bipartitheit soll der Algorithmus eine mögliche Unterteilung V_1, V_2 ausgeben, sonst *nicht bipartit*. Geben Sie **Pseudocode** an.
3. Erweitern Sie ihren Algorithmus, so dass im Falle der Nichtbipartitheit ein Zeuge ausgegeben wird. **Hinweis:** Pseudocode ist nicht zwingend erforderlich.

Aufgabe 3 (Dichte und spärliche Graphen, 2 + 2 + 2 + 2 + 2 Punkte)

Eine **Familie von Graphen** ist eine unendliche Menge von Graphen, mit der Eigenschaft dass sich für jedes $n_0 \in \mathbb{N}_{\geq 0}$ stets ein Graph mit $n \geq n_0$ Knoten in der Menge befindet.

1. Zeigen Sie: Vollständige ungerichtete Graphen mit n Knoten haben $\Theta(n^2)$ Kanten.

Eine Familie von ungerichteten Graphen heie eine Familie **spärlicher** Graphen, wenn jeder Graph mit n Knoten $O(n)$ Kanten hat.

2. Zeigen Sie: In einer Familie spärlicher Graphen liegt der maximale Knotengrad nicht notwendigerweise in $O(1)$.

Eine Familie von ungerichteten Graphen heie eine Familie **gleichmäßig spärlicher** Graphen, wenn jeder Knoten jedes Graphen mit n Knoten $O(1)$ Kanten hat.

3. Zeigen Sie: Eine Familie gleichmäßig spärlicher Graphen ist auch eine Familie spärlicher Graphen.

Eine Familie von ungerichteten Graphen heie eine Familie **dichter** Graphen, wenn jeder Graph mit n Knoten $\Omega(n^2)$ Kanten hat.

4. Zeigen Sie: In einer Familie dichter Graphen hat nicht unbedingt jeder Knoten eines Graphen mit n Knoten $\Omega(n)$ Kanten.

Eine Familie von ungerichteten Graphen heie eine Familie **gleichmäßig dichter** Graphen, wenn jeder Knoten jedes Graphen mit n Knoten $\Omega(n)$ Kanten hat.

5. Zeigen Sie: Eine Familie gleichmäßig dichter Graphen ist auch eine Familie dichter Graphen.

Aufgabe P9 (Maximum Independent Set, optional)

Für die praktischen Übungen verwenden wir die Plattform www.hackerrank.com. Hier müssen Sie sich registrieren um an den Übungen teilzunehmen. Unter dem Link

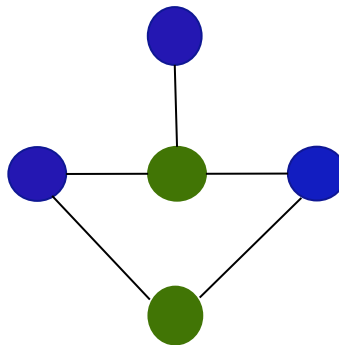
<https://www.hackerrank.com/adsi-2023>

finden die praktischen Übungen in der Form eines Programmierwettbewerbs statt.

In der neunten Challenge geht es um das Problem, eine möglichst große, maximale stabile Menge zu berechnen. Für einen gegebenen Graphen $G = (V, E)$ bezeichnet man eine Teilmenge S von V als stabile Menge (*engl. independent set*), falls die Knoten in S nicht zueinander adjazent sind. Eine maximale stabile Menge ist eine stabile Menge S' , sodass für jeden Knoten $u \in V \setminus S'$, $S' \cup \{u\}$ keine stabile Menge mehr ist.

Ihnen wird ein ungerichteter, ungewichteter Graph G als Eingabe gegeben. Ihre Aufgabe ist es, eine Heuristik zu entwerfen, welche eine möglichst große, maximale stabile Menge findet.

Beachten sie, es gibt mehrere maximale stabile Mengen, welche unterschiedlich groß sein können, wie das folgende Beispiel zeigt:



In diesem Graphen bilden sowohl die grünen, als auch die blauen Knoten maximale stabile Mengen.

Auf Moodle finden Sie wieder ein Framework für diese Aufgabe: *framework9.cpp*. Der Score für diese Aufgabe wird nicht mittels der Laufzeit Ihres Algorithmus berechnet, sondern wird über die Größe der maximalen stabilen Menge berechnet, welche Sie finden.