Technische Universität Wien
Institut für Computergraphik und Algorithmen
Algorithms and Complexity Group

186.815 Algorithmen und Datenstrukturen 2 VU 3.0

1. Test SS 2017

29. Juni 2017

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: ______ Vorname: ______

Matrikelnummer: ______ Unterschrift: ______

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.

Sie dürfen die Lösungen nur auf die Angabeblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte!).

Die Verwendung von Taschenrechnern, Mobiltelefonen, Tablets, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

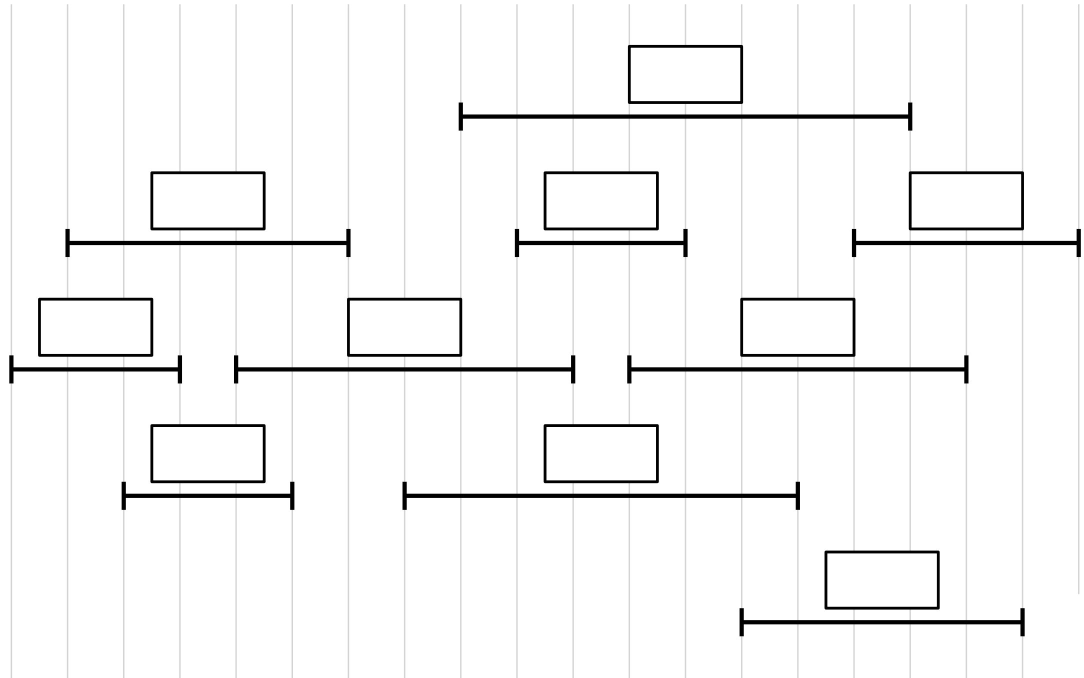
	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	16	16	18	50
Erreichte Punkte:				

Viel Erfolg!

Aufgabe A1: Reduktionen und NP-Spezialfälle **(16 Punkte)**

a) (8 Punkte)

Gegeben sei die untenstehende Intervallrepräsentation eines Graphen $G = (V, E)$. Führen Sie den aus der Vorlesung bekannten Algorithmus zum Färben von Intervallgraphen mit einer minimalen Anzahl von Farben $\{0, 1, 2, \dots\}$ aus. Schreiben Sie an jedes Intervall dessen Farbe.



b) (8 Punkte)

Seien A,B,C Ja/Nein-Probleme in NP und n die Eingabegröße. Nehmen Sie an, es gibt

- eine Reduktion von A nach B in Zeit $O(n^3)$,
- eine Reduktion von B nach C in Zeit $O(n!)$,
- eine Reduktion von C nach A in Zeit $O(n^2)$.

1) Nehmen Sie an, es existiert ein Algorithmus, der Problem B in Zeit $O(n^4)$ löst. Was können Sie dann über die anderen Probleme aussagen? Kreuzen Sie die richtigen Antworten an.

B in $O(n^4)$	in P	NP-vollständig	keine Aussage möglich
A ist			
C ist			

2) Nehmen Sie an, A sei NP-vollständig. Was können Sie dann über die anderen Probleme aussagen? Kreuzen Sie die richtigen Antworten an.

A NP-vollständig	in P	NP-vollständig	keine Aussage möglich
B ist			
C ist			

Aufgabe A2: Branch-and-Bound (16 Punkte)

a) (8 Punkte)

Betrachten Sie folgenden Pseudocode eines generischen Branch-and-Bound-Algorithmus für ein **Minimierungsproblem**.

`Branch-and-Bound-Min(`I`):`

$A \leftarrow$ `Wert einer initialen heuristischen Lösung`
$\Pi \leftarrow \{I\}$

`while` $\exists I' \in \Pi$
    `Entferne` I' `aus` Π
    `Berechne für` I' `lokale` $\boxed{\qquad\qquad\qquad}$ `Schranke` B'
    `if` A $\boxed{\quad}$ B'
        $A' \leftarrow$ `Wert einer heuristischen Lösung für` I'
        `if` A' `entspricht einer gültigen Lösung für` I'
            `if` $A' < A$
                $\boxed{\qquad\qquad\qquad\qquad\qquad}$
        `if` B' $\boxed{\quad}$ A
            `Partitioniere` I' `in Teilinstanzen` $I_1, \ldots, I_k$
            $\Pi \leftarrow \Pi \cup \{I_1, \ldots, I_k\}$
`return beste gefundene Lösung mit Wert` A

Füllen Sie die vier Lücken im Pseudocode so aus, dass ein korrekter Branch-and-Bound-Algorithmus entsteht.

b) (2 Punkte)

Beschreiben Sie in 1–2 Sätzen die Best-First-Strategie zur Auswahl des nächsten zu bearbeitenden Teilproblems in einem Branch-and-Bound Algorithmus.

c) (6 Punkte)

Ist der folgende Algorithmus geeignet, um eine untere Schranke in einem Branch-and-Bound-Algorithmus zum Finden eines minimalen Vertex Cover in einem Graphen $G = (V, E)$ zu berechnen? Begründen Sie Ihre Antwort (falls Ihre Antwort 'JA' ist, begründen Sie warum die Schranke *immer* gilt, falls Ihre Antwort 'NEIN' ist, erklären Sie warum die Schranke *nicht immer* gilt).

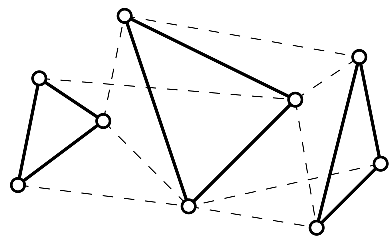
```
Maybe-Lower-Bound(G = (V, E)):

bound ← 0
while E ≠ ∅
    (u, v) ← beliebige Kante aus E
    if ∃w ∈ V mit (u, w), (v, w) ∈ E
        bound ← bound+3
        entferne u, v, w aus V und alle zu u, v, w inzidenten Kanten aus E
    else
        bound ← bound+1
        entferne u, v aus V und alle zu u, v inzidenten Kanten aus E
return bound
```

Aufgabe A3: Approximation **(18 Punkte)**

Betrachten Sie das aus der Vorlesung bekannte Minimierungsproblem Kleinstes Vertex Cover für einen Graphen $G = (V, E)$ mit $|V| = 3n$ Knoten. Nehmen Sie an, G lässt sich in ein perfektes Dreiecksmatching zerteilen. Das bedeutet, es gibt eine Menge $T = \{\Delta_1, \ldots, \Delta_n\}$ von n knotendisjunkten Dreiecken, wobei jedes $\Delta_i = (V_i, E_i)$ ein Teilgraph von G mit drei Knoten und drei Kanten ist, d.h. $V_i = \{v_1^i, v_2^i, v_3^i\}$ und $E_i = \{(v_1^i, v_2^i), (v_2^i, v_3^i), (v_3^i, v_1^i)\}$. Weiters gilt $V = \bigcup_{i=1}^{n} V_i$.

Die Abbildung zeigt ein Beispiel eines perfekten Dreiecksmatchings (dicke Kanten) in einem Graphen mit 9 Knoten.



a) (3 Punkte)

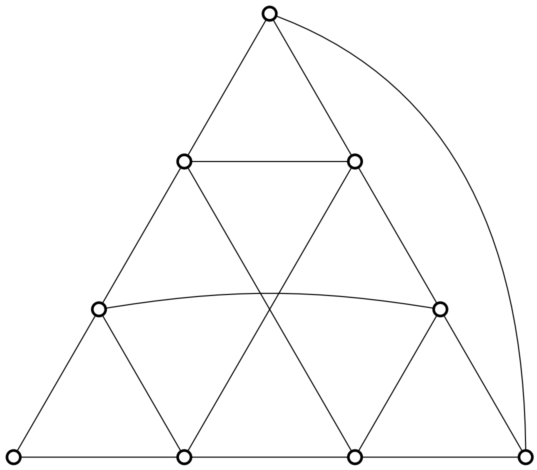
Der vollständige Graph K_n mit n Knoten ist ein Graph, in dem alle Knoten paarweise durch eine Kante verbunden sind. Lässt sich der vollständige Graph K_{12} in ein perfektes Dreiecksmatching zerteilen? Begründen Sie kurz Ihre Antwort.

b) (3 Punkte)

Wie viele Knoten enthält ein kleinstes Vertex Cover in dem Graphen K_{12}?

c) (6 Punkte)

Unten sehen Sie einen Graphen mit 9 Knoten und 16 Kanten. Können Sie 7 Kanten hinzufügen, so dass der entstehende Graph einfach ist und ein Vertex Cover der Größe 6 enthält? Zeichnen Sie, falls möglich, die 7 Kanten ein und markieren Sie ein Vertex Cover der Größe 6. Andernfalls zeichnen Sie weniger als 7 Kanten ein und markieren Sie ein Vertex Cover der Größe 6.



d) (6 Punkte)

Die komplette Knotenmenge V ist nach Definition immer ein Vertex Cover. Welche bestmögliche Gütegarantie erfüllt die Menge V für das Problem KLEINSTES VERTEX COVER für einen Graphen G, der sich in ein perfektes Dreiecksmatching zerlegen lässt? Beweisen Sie Ihre Antwort. Betrachten Sie dazu die Lösungsgröße c_{opt} eines kleinsten Vertex Covers von G und vergleichen Sie c_{opt} mit $|V|$.