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

186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1.Übungstest SS 2016

28. April 2016

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:	20	20	10	50
Erreichte Punkte:				

Viel Erfolg!

Aufgabe A1: Algorithmenanalyse **(20 Punkte)**

a) (10 Punkte) Tragen Sie für die Codestücke `FunktionA` und `FunktionB` jeweils die Laufzeit und den Rückgabewert (z) in Abhängigkeit von n in Θ-Notation in die nachfolgende Tabelle ein.

	`FunktionA`	`FunktionB`
Laufzeit		
Rückgabewert (z)		

`FunktionA`(n):
$x \leftarrow 50000$
while $x > 1$
  for $j \leftarrow 1$ `bis` $\left\lfloor \frac{n}{50} \right\rfloor$
    $z \leftarrow 2j$
  $x \leftarrow \frac{x}{5}$
return z

`FunktionB`(n):
$x \leftarrow 1$
$y \leftarrow 0$
$z \leftarrow 0$
while $x < n^2$
  while $y \leq x$
    $y \leftarrow y + 1$
    $z \leftarrow z + n$
  $x \leftarrow x + 1$
return z

b) (6 Punkte) Beantworten Sie die nachfolgenden Fragen und begründen Sie jeweils Ihre Antwort in **wenigen** Worten!

- Wenn die Worst-Case-Laufzeit eines Algorithmus in $\Theta(n^2)$ liegt, ist es dann möglich, dass seine Laufzeit für **manche** Instanzen in $O(n)$ liegt?

- Wenn die Worst-Case-Laufzeit eines Algorithmus in $\Theta(n^2)$ liegt, ist es dann möglich, dass seine Laufzeit für **alle** Instanzen in $O(n)$ liegt?

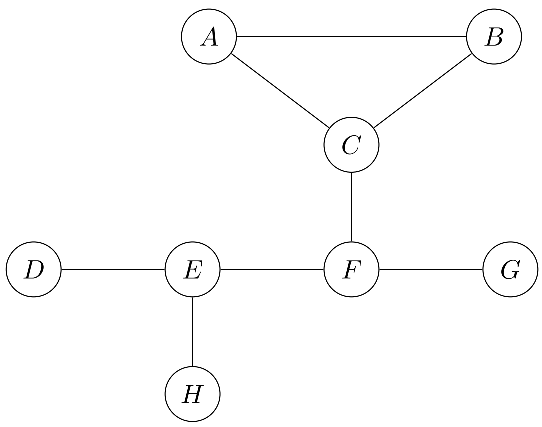
c) (4 Punkte) Ordnen Sie folgende Funktionen nach Dominanz, beginnend mit der asymptotisch am schwächsten wachsenden. Es genügt die Funktionen zu reihen, ein Beweis der Gültigkeit der Relationen ist nicht erforderlich.

$$\log\left(n^{15}\right), \quad \left(\frac{3}{2}\right)^n, \quad n - n^3 + 7n^5, \quad \left(\frac{1}{3}\right)^n, \quad \sqrt{n^8}, \quad n^2\left(\log n\right)^2$$

Aufgabe A2: Graphen **(20 Punkte)**

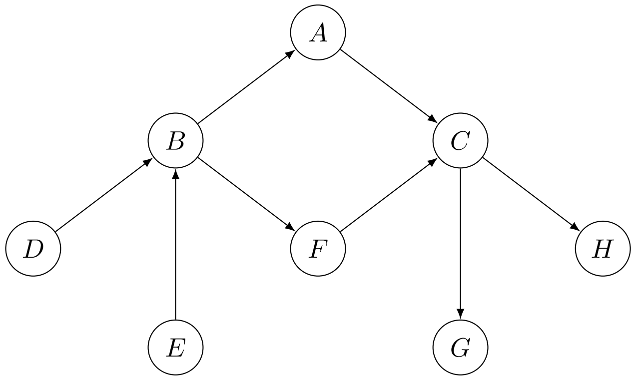
a) (12 Punkte) Betrachten Sie den nachfolgenden Graphen $G = (V, E)$. Gehen Sie alle Knoten $v \in V$ durch und überlegen Sie jeweils, ob es möglich ist, dass sowohl die Breiten- als auch die Tiefensuche (jeweils gemäß dem Pseudocode aus den Vorlesungsfolien) mit v als Startknoten den Graphen in derselben Reihenfolge abarbeiten.

- Ist das für den jeweils betrachteten Knoten v **möglich**, dann geben Sie eine passende Abarbeitungsreihenfolge an.
- Ist das für den jeweils betrachteten Knoten v **nicht möglich**, dann soll dies stattdessen durch Aufteilen der Knoten in zwei Mengen X und Y ($X \cup Y = V, X \cap Y = \emptyset$) bewiesen werden:
  - Für die Breitensuche soll gelten, dass immer alle Knoten aus X vor allen Knoten aus Y berücksichtigt werden.
  - Für die Tiefensuche muss es aber immer ein Knotenpaar $x \in X$ und $y \in Y$ geben, sodass y vor x besucht wird.



Knoten	Abarbeitungsreihenfolge	X	Y
A			
B			
C			
D			
E			
F			
G			
H			

b) (8 Punkte) Gegeben sei der folgende gerichtete Graph:



- Finden Sie für diesen Graphen eine topologische Sortierung. (2 Punkte)

- Wie viele unterschiedliche topologische Sortierungen gibt es für diesen Graphen? (3 Punkte)

- Zeichnen Sie **eine** zusätzliche Kante ein, sodass keine gültige topologische Sortierung mehr möglich ist. Begründen Sie in einem Satz, warum das so ist. (3 Punkte)

Aufgabe A3: Sortieren **(10 Punkte)**

a) (2 Punkte) Das Array $A = [15, 18, 6, 2, 9, 11]$ wird mittels Selection-Sort gemäß dem Pseudocode aus den Vorlesungsfolien sortiert. Kreuzen Sie in nachfolgender Liste jene Arrays an, die nach einer oder mehreren Iterationen der äußersten Schleife entstehen können.

☐ $A = [2, 6, 18, 15, 9, 11]$
☐ $A = [6, 15, 18, 2, 9, 11]$
☐ $A = [11, 15, 18, 6, 2, 9]$
☐ $A = [2, 6, 9, 11, 18, 15]$

b) (8 Punkte) Im Folgenden seien vier Arrays A, B, C, D mit jeweils n Elementen gegeben:

- A: aufsteigend sortiert
- B: absteigend sortiert
- C: jedes Element an einer ungeraden Position ist kleiner als jedes Element an einer geraden Position (z.B. $5, 4, 6, 2, 8, 1, 7, 3$)
- D: ausgehend von einem aufsteigend sortierten Array wird jedes Element an gerader Position mit dem direkt darauffolgenden Element ungerader Position vertauscht (z.B. $2, 1, 4, 3, 6, 5, 8, 7$)

Geben Sie für jedes Array die Laufzeit von Insertion-Sort (gemäß dem Pseudocode aus den Vorlesungsfolien) in Θ-Notation an:

Array	$\Theta(\cdot)$
A	
B	
C	
D	

Nehmen Sie an, dass Arrays immer mit 0 beginnend indiziert sind, 0 gilt als gerade.