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

186.815 Algorithmen und Datenstrukturen 2 VU 3.0
Übungstest SS 2015
25. Juni 2015

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: ______ Vorname: ______

Matrikelnummer: ______ Unterschrift: ______

Anzahl abgegebener Zusatzblätter: ______

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!).

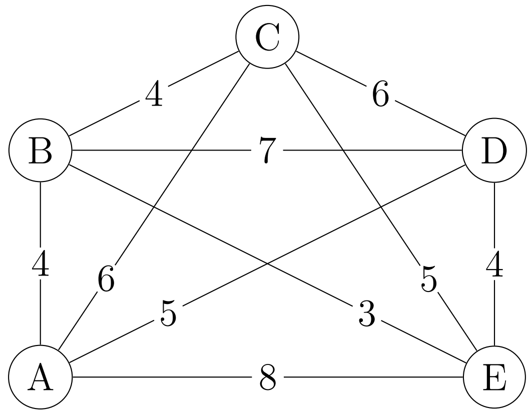
Die Verwendung von Taschenrechnern, Mobiltelefonen, Tablets, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

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

Viel Erfolg!

Aufgabe 1.A: Approximationsalgorithmen **(18 Punkte)**

Gegeben ist der folgende, vollständige, ungerichtete Graph G von fünf Städten und den jeweiligen Distanzen. Wenden Sie die Christophides-Heuristik an, um eine Näherungslösung für das Traveling-Salesman-Problem in diesem Graphen zu finden.



a) (3 Punkte)

Markieren Sie in G jene Kanten, die den minimalen Spannbaum T bilden, und geben Sie die Menge W der Knoten mit ungeradem Grad in T an.

b) (4 Punkt)

Geben Sie das perfekte Matching M kleinsten Gewichts im von W induzierten Graphen an und zeichnen Sie den Graphen G' mit den Kanten $T \cup M$.

c) (4 Punkte)

Zeichnen Sie eine Euler-Tour F im Graphen G'. Beginnen Sie beim Knoten A und orientieren Sie F, indem Sie, falls mehrere Knoten in Frage kommen, immer zu dem Knoten mit dem alphabetisch erstgereihten Buchstaben weitergehen.

d) (3 Punkte)

Zeichnen Sie die Ergebnistour der Christophides-Heuristik.

e) (4 Punkte)

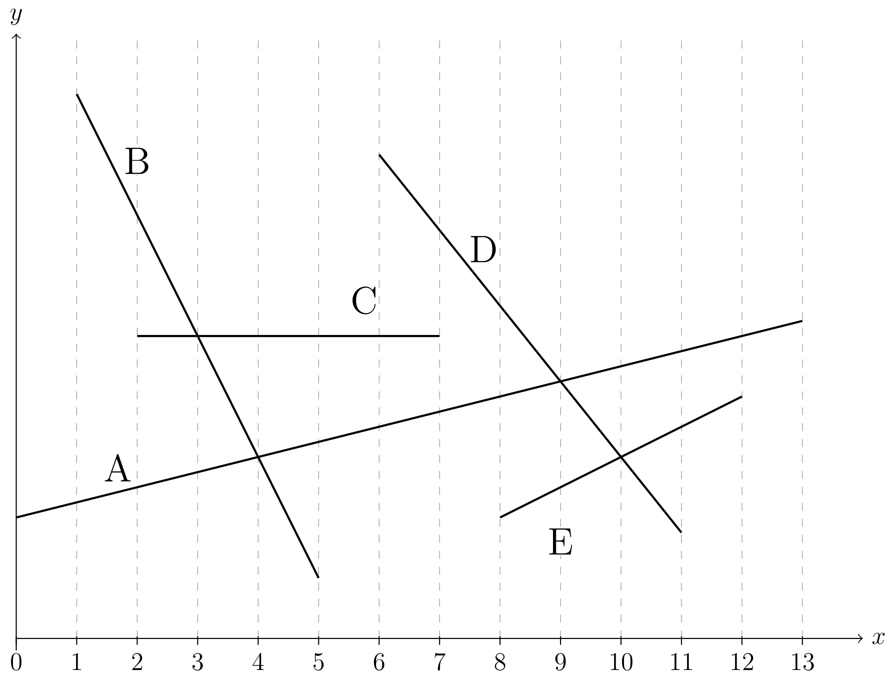
Welche Gütegarantie besitzt die *Christophides-Heuristik* für das metrische Traveling-Salesman-Problem? Beschreiben Sie die Bedeutung dieses Faktors auch in Worten.

Platz für Notizen

Aufgabe 2.A: Geometrische und Approximative Algorithmen (16 Punkte)

a) **Schnitt von allgemeinen Liniensegmenten (8 Punkte)**

Führen Sie den aus der Vorlesung bekannten *Scan-Line Algorithmus zum Schnitt von allgemeinen Liniensegmenten* aus.
(Die vertikalen Linien markieren die Zeitpunkte und dienen zur Orientierung.)



i) (2 Punkte)

Geben Sie dabei für die folgenden Zeitpunkte den Zustand der Scan-Line-Status Struktur an, nachdem das jeweilige Ereignis abgearbeitet wurde.

- Zeitpunkt 2:
- Zeitpunkt 9:

ii) (4 Punkte)

Geben Sie die Zeitpunkte an, wann die Schnittpunkte in die Ereignisstruktur eingefügt werden.

- $A \cap B$:
- $D \cap E$:

iii) (2 Punkte)

Welche Operationen bzw. Anweisungen werden beim *Scan-Line Algorithmus zum Schnitt von allgemeinen Liniensegmenten* ausgeführt, wenn das Ereignis „Segment Ende“ abgearbeitet wird?

b) **Approximative Algorithmen (8 Punkte)**

Graph Coloring: Gegeben sei ein ungerichteter Graph $G = (V, E)$ und eine geordnete Menge von Farben $C = \{c_1, c_2, \dots\}$. Ziel ist es nun, den Graph mit **möglichst wenigen unterschiedlichen** Farben so zu färben, dass zwei adjazente Knoten niemals dieselbe Farbe haben, d.h., den beiden Endpunkte einer jeden Kante müssen immer verschiedene Farben zugewiesen werden.

Algorithmus 1 colorGreedy(V, E, C)

---

solange $\exists$ ein ungefärbter Knoten **{**
  wähle einen ungefärbten Knoten $v \in V$ mit maximalem Knotengrad;
  weise v die erste Farbe aus C zu, sodass kein adjazenter Knoten dieselbe Farbe hat;
}

Hinweis: Der Grad eines Knotens $v \in V$ ist $d(v) = |\{u | \{u, v\} \in E, u \neq v\}|$

i) (6 Punkte)

Zeigen Sie anhand eines Beispiels, dass der Algorithmus `colorGreedy` nicht für jeden Graphen eine Färbung mit der minimalen Anzahl an benötigten Farben findet. Stellen sie die einzelnen Schritte dar, die der Algorithmus zur Berechnung dieser suboptimalen Lösung ausführt.

Geben Sie außerdem eine Lösung an, die weniger Farben benötigt als das Ergebnis von `colorGreedy`.

ii) (2 Punkte)

Wie viele Farben benötigt der Algorithmus `colorGreedy` höchstens, um einen Graphen $G = (V, E)$ zu färben. Verwenden Sie für die Abschätzung die Knotengrade des Graphen.

Aufgabe 3.A: Amortisierte Analyse **(16 Punkte)**

Gegeben sei eine Datenstruktur Stack mit folgenden Operationen:

- **Push**: Lege ein Element auf den Stack.
- **Empty&push**: Entferne Elemente vom Stack (*pop*), bis keine Elemente mehr am Stack liegen und führe dann eine push-Operation aus.

  (Hinweis: Wenn also k Elemente am Stack liegen, wird zuerst k-mal *pop* ausgeführt und danach einmal *push*)

Nehmen Sie an, dass die benötigte Zeit für das Einfügen (*push*) und das Entfernen (*pop*) eines einzelnen Elements jeweils 1 ist.

a) (2 Punkte)

Geben Sie die Worst-Case-Laufzeit für *push* und *empty&push* an, wenn davor bereits n (beliebige) Operationen auf einem anfangs leeren Stack ausgeführt wurden.

b) (8 Punkte)

Führen Sie eine **amortisierte Analyse** für eine Sequenz von n Operationen auf einem anfangs leeren Stack aus und zeigen Sie mittels **Potentialfunktion-Methode**, dass die amortisierte Laufzeit von *push* und *empty&push* konstant ist.

c) (6 Punkte)

Wodurch unterscheidet sich die amortisierte Analyse von der herkömmlichen Analyse? Unter welchen Voraussetzungen bringt die amortisierte Analyse Vorteile?