ALP III: Datenstrukturen und Datenabstraktion
13. Aufgabenblatt
Übungsgruppe 1.8: Marcel Erhardt

Tobias Lohse/ Marvin Kleinert/ Anton Drewing

30.01.2015

Aufgabe 1

(a) 3COLORING $\in$ NP
gegeben: ungerichteter Graph G, dargestellt durch Adjazenzliste
gesucht: Färbung der Knoten von G, sodass jede Kante zwei unterschiedlich farbige Knoten verbindet
Zertifiakt c: Liste mit Farben der Knoten
Verifikator A:

```
A(G, c):
  for edge in G:
    if c[edge.fst] == c[edge.snd]:
      return false;
  return true;
```

Laufzeit von A: $O(n^2)$ mit $n = |V|$

(b) COMPOSITE $\in$ NP
gegeben: natürliche Zahl k in Binärdarstellung
gesucht: ganzzahliger Teiler von k
Zertifikat c: ganzzahliger Teiler von k
Verifikator A:

```
A(k, c):
  return (k mod c == 0);
```

Laufzeit: $O(n^2)$

(c) SAMECOMPONENTS $\in$ NP
gegeben: ungerichteter Graph $G = (V, E)$ und zwei Knoten $u, v \in V$
gesucht: Zusammenhangskomponente von G, in der u und v liegen
Zertifikat: Knotenliste von G, in der u und v liegen
Verifikator A:

```
A(<G, u, v>, c):
  uInC, vInC = false;
  for vertex in c:
    if u == vertex:
      uInC = true;
    if v == vertex:
      uInC = true;
  return (uInC && vInC);
```

Laufzeit: O(n) mit $n = |V|$

Aufgabe 2

Java-Code: siehe E-Mail-Abgabe

Aufgabe 3

(a) SAMESUM $\in$ NP-vollständig

1. SAMESUM $\in$ NP
   gegeben: Folge $setA = a_1, ..., a_n$ natürlicher Zahlen
   gesucht: zwei Teilfolgen mit gleich großen Summen
   Zertifikat c: zwei Teilfolgen $< c1, c2 >$
   Verifikator A:

```
A(setA,<c1,c2>):
  sum1, sum2 = 0;
  for number in c1:
    sum1 += number;
  for number in c2:
    sum2 += number;
  return (sum1 == sum2);
```

   Laufzeit: O(n)

2. SUBSET-SUM $\leq_p$ SAMESUM
   Funktion f: $f(a_1, ..., a_n, b) = (a_1, ..., a_n, s + b, 2s - b)$
   $w \in SUBSET - SUM \Rightarrow f(w) \in SAMESUM$:
   Wenn w in SUBSET-SUM liegt, gibt es eine Teilfolge von $a_1, ..., a_n$, die summiert b ergibt. Diese bildet in f(w) mit 2s-b eine Menge, deren Summe 2s ist. Die übrigen Elemente der Folge ergeben summiert s-b, was zusammen mit dem Element s+b ebenfalls 2s ist.
   $w \in SUBSET - SUM \Leftarrow f(w) \in SAMESUM$:
   Wenn f(w) in SAMESUM liegt, muss eine Teilfolge von $a_1, ..., a_n$ existieren , die b ergibt, da andernfalls eine Summe von Elementen aus f(w) größer und eine Summe kleiner als 2s ist.

(b) VERTEX-COVER $\in$ NP-vollständig

1. VERTEX-COVER $\in$ NP
   gegeben: ungerichteter Graph $G = (V, E)$ und Zahl $k \in N$
   gesucht: Menge $V' \subseteq V$, sodass jede Kante in E zu mind. einem Knoten in V' inzident ist
   Zertifikat c: Menge V' mit Größe k
   Verifikator A:

```
A(<(V,E),k>,c):
  if c.size != k:
    return false;
  for edge in E:
    if !c.contains(edge.fst) && !c.contains(edge.snd):
      return false;
  return true;
```

   Laufzeit: $(n - 1) * n/2 * n = O(n^3)$ mit $n = |V|$

2. CLIQUE $\leq_p$ VERTEX-COVER
   Funktion f: $f(G, k) = (\overline{G}, n - k)$
   $w \in CLIQUE \Rightarrow f(w) \in VERTEX - COVER$:

Wenn w in CLIQUE liegt, hat G eine Clique mit k Knoten. Im Komplementärgraphen wählt man als V' die (n-k) Knoten, die nicht in der Clique sind, sodass alle Kanten in $\overline{G}$ mindestens zu einem Knoten aus V' inzident sind.
$w \in CLIQUE \Leftarrow f(w) \in VERTEX-COVER$:
Wenn f(w) in VERTEX-COVER liegt, gibt es eine Menge V' mit (n-k) Knoten, sodass jede Kante in $\overline{E}$ zu mindestens einem Knoten in V' inzident ist. Zwischen den k Knoten, die nicht in V' liegen, darf es demnach keine Kanten untereinander geben, was im eigentlichen Graphen G dann einen vollständigen Graphen mit k Knoten, also eine k-Clique zur Folge hat.