Wintersemester 2018/19
Professor Dr. S. Hougardy
Dr. U. Brenner

Forschungsinstitut für Diskrete Mathematik
Universität Bonn

Algorithmische Mathematik I
8. Übung

1. Es sei G ein zusammenhängender einfacher ungerichteter Graph mit mehr als einem Knoten. Zeigen Sie, dass G einen Knoten v enthält mit

$$\frac{1}{|\delta(v)|} \sum_{w \in N(v)} |\delta(w)| \;\geq\; \frac{2|E(G)|}{|V(G)|}.$$

(4 Punkte)

2. Es sei G ein einfacher ungerichteter Graph. Zeigen Sie, dass es

   (a) zwei disjunkte Mengen A und B mit $A \cup B = V(G)$ gibt, so dass in $G[A]$ und in $G[B]$ alle Knoten geraden Grad haben.

   (b) zwei disjunkte Mengen C und D mit $C \cup D = V(G)$ gibt, so dass in $G[C]$ alle Knoten geraden Grad und in $G[D]$ alle Knoten ungeraden Grad haben. (6 Punkte)

3. Beweisen Sie:

   (a) Seien (V, F_1) und (V, F_2) zwei Branchings mit $2|F_1| < |F_2|$. Dann gibt es eine Kante $e \in F_2 \setminus F_1$, so dass $(V, F_1 \cup \{e\})$ ein Branching ist.

   (b) Die Aussage aus (a) wird falsch, wenn man die Bedingung „$2|F_1| < |F_2|$“ durch „$2|F_1| \leq |F_2|$“ ersetzt. (3+2 Punkte)

4. Schreiben Sie eine Klasse zum Rechnen mit Polynomen in einer Variablen x mit Koeffizienten in $\mathbb{Z}$, d.h. für Polynome der Form $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x^1 + a_0$ mit $a_i \in \mathbb{Z}$ für $i \in \{0, \ldots, n\}$.

   Vervollständigen Sie dazu den folgenden Rahmen für die Klasse:

```
1  // Polynomial.h (class for polynomials)
2
3  #include <vector>
4  #include <string>
5
6  class Polynomial{
7
8  friend Polynomial operator*(int lhs, const Polynomial & rhs);
9
10 public:
11     Polynomial();                                  // construct zero polynomial
12     void SetCoefficient(size_t index, int value); // set coefficient for x^index
13     std::string toString() const;                  // convert to string
14
15 private:
16     std::vector<int> _coefficient;
17 };
18
19 Polynomial operator*(int lhs, const Polynomial & rhs) {
20     // scalar multiplication
21 }
```

Das Schlüsselwort `friend` erlaubt dabei dem neu zu definierenden Operator `*` den Zugriff auf die privaten Members der Klasse `Polynomial`.

Hilfreich für Ihre Implementation könnten darüber hinaus die Funktionen `std::abs()` und `std::to_string()` sein, die den Absolutwert einer Zahl zurückliefern bzw. eine Zahl in einen `string` konvertieren.

Testen Sie Ihre Klasse mit dem folgenden Programm:

```
1  // Test program for Polynomial.h
2
3  #include <iostream>
4  #include "Polynomial.h"
5
6  int main() {
7      Polynomial P;
8      P.SetCoefficient(7, 1);
9      P.SetCoefficient(3, 6);
10     P.SetCoefficient(0,-1);
11     std::cout << "     P = " <<        P.toString() << "\n";
12     std::cout << "-1 * P = " << (-1 * P).toString() << "\n";
13     std::cout << " 5 * P = " << ( 5 * P).toString() << "\n";
14     std::cout << "-5 * P = " << (-5 * P).toString() << "\n";
15     std::cout << " 0 * P = " << ( 0 * P).toString() << "\n";
16 }
```

Die Ausgabe des Programms muss exakt (Leerzeichen beachten!) wie folgt aussehen:

```
     P = x^7 + 6 x^3 - 1
-1 * P = - x^7 - 6 x^3 + 1
 5 * P = 5 x^7 + 30 x^3 - 5
-5 * P = - 5 x^7 - 30 x^3 + 5
 0 * P = 0
```

(5 Punkte)

Abgabe: Montag, den 3.12.2018, bis 10:12 Uhr.

Abgabe der Programmierübungen:
Per E-Mail an `alma_prog_gr_XX@dm.uni-bonn.de`, wobei `XX` durch die Nummer Ihrer Übungsgruppe zu ersetzen ist, also z.B. `alma_prog_gr_07@dm.uni-bonn.de`, wenn Sie in Gruppe 7 sind, oder `alma_prog_gr_12@dm.uni-bonn.de`, wenn Sie in Gruppe 12 sind. Wenn Sie Ihre Übungsgruppe nicht kennen, schreiben Sie an `alma_prog_gr_unbekannt@dm.uni-bonn.de`.

Öffnungszeiten des Help Desks:
Montags, 16 – 19 Uhr und freitags, 12 – 15 Uhr, jeweils in Raum N1.002, Endenicher Allee 60, Nebengebäude.
`www.mathematics.uni-bonn.de/files/bachelor/help-desk.pdf`

Zusätzlich gibt es ab sofort einen **Help Desk für Programmierfragen**, und zwar immer freitags, 8 – 10 Uhr und 12 – 16 Uhr, im PC-Pool in der Wegelerstraße 6, Raum E02 (Hochschulrechenzentrum).