

Aufgabe 1: Superstar

1.1 Lösungsidee

Wenn wir die Anfrage „Folgt Mitglied x Mitglied y “ an das soziale Netzwerk stellen, können wir aus der Antwort eines der beiden Mitglieder als Superstar ausschließen: Falls die Antwort „Ja“ ist, kann x kein Superstar sein, weil x jemandem folgt. Falls die Antwort „Nein“ ist, kann y kein Superstar sein, weil nicht alle Mitglieder y folgen.

Dies können wir uns zunutze machen, um alle bis auf ein Mitglied als Superstar auszuschließen. Falls das Netzwerk n Mitglieder hat, brauchen wir dafür $(n - 1)$ -Fragen, die noch nicht ausgeschlossene Mitglieder beinhalten. So verwenden wir für die verbliebenen Kandidaten folgende Suchfunktion *Suche-Superstar-Kandidat*:

```

1 function Suche-Superstar-Kandidat
2   Input: Mitglieder
3   Output: Superstar-Kandidat
4 Kandidaten-Liste = liste(Mitglieder)
5 while |Kandidaten-Liste| > 1 do
6     x = erstes Mitglied der Kandidaten-Liste
7     y = zweites Mitglied der Kandidaten-Liste
8     if(folgt(x,y)) then
9         entferne x aus der Kandidaten-Liste
10    else
11        entferne y aus der Kandidaten-Liste
12    end
13 end
14 return Mitglied, das bis zuletzt in der Kandidaten-Liste verblieb

```

Falls das Netzwerk einen Superstar hat, so ist er auf jeden Fall das von dieser Suchfunktion zurückgegebene Mitglied, weil ein Superstar niemals von dem in der Funktion angewandten Verfahren aus der Liste der Kandidaten entfernt werden kann. Da in jedem Schleifendurchlauf ein Kandidat entfernt wird, terminiert die Schleife.

Um sicherzustellen, dass wir tatsächlich einen Superstar gefunden haben, müssen wir noch die Bedingung überprüfen, ob jedes andere Mitglied dem Superstar-Kandidaten folgt und der Superstar-Kandidat keinem anderen Mitglied folgt. Dies können wir wie folgt in der Prüffunktion *Überprüfe-Superstar-Kandidaten* bestimmen:

```

1 function Überprüfe-Superstar-Kandidaten
2   Input: Superstar-Kandidat, Mitglieder
3   Output: Superstar-Kandidat oder "Kein Superstar gefunden"
4 foreach x in Mitglieder \ Superstar-Kandidat do
5     if(folgt(Superstar-Kandidat, x) or not(folgt(x, Superstar-Kandidat)))
6         then
7             return "Kein Superstar gefunden"
8         end
9 end
10 return Superstar-Kandidat

```

Für diese Überprüfung des Superstar-Kandidaten benötigen wir $2(n - 1)$ weitere Fragen. Insgesamt müssen also $3(n - 1)$ Fragen an das Netzwerk gestellt werden.

1.2 Verringerung der Anfragenzahl

Beim Überprüfen wiederholen wir Fragen, für die wir bereits bei der Suche eine Antwort erhalten haben. Wenn wir uns diese Antworten merken, können wir Anfragen sparen.

Dies lässt sich sogar noch weiter ausnutzen, indem wir bei der Suche die Anfragen möglichst gleichmäßig auf die Kandidaten verteilen. So ist sichergestellt, dass wir möglichst viel über unseren Kandidaten erfahren. Hierfür ist eine kleine Anpassung unseres Suchverfahrens notwendig.

Wir entfernen in jedem Schritt statt einen Kandidaten die beiden aktuellen Kandidaten aus der Liste und fügen den einen der beiden, der weiter Superstar sein könnte, hinten wieder an die Liste an (diese Datenstruktur nennt man auch Warteschlange bzw. Queue nach dem FIFO-Prinzip). So stellen wir sicher, dass kein Mitglied, das kein Superstar ist, aber erst spät während der Suche ausgesondert wird, viele Anfragen über sich verbraucht:

```

1 function Verbesserte-Suche-Superstar-Kandidat
2   Input: Mitglieder
3   Output: Superstar-Kandidat
4   Kandidaten-Liste = liste(Mitglieder)
5   while |Kandidaten-Liste| > 1 do
6     x = entferne erstes Mitglied von Kandidaten-Liste
7     y = entferne erstes Mitglied von Kandidaten-Liste
8     if(folgt(x,y)) then
9       hänge y hinten an Kandidaten-Liste an
10    else
11      hänge x hinten an Kandidaten-Liste an
12    end
13  end
14  return Mitglied, das bis zuletzt in der Kandidaten-Liste verblieb

```

1.3 Beispiele

Für die online angegebenen Beispiele könnte das eben verbesserte Verfahren wie folgt ablaufen:

superstar1.txt

```

1 [1] Beantwortete Anfrage: Selena folgt Justin
2 [2] Beantwortete Anfrage: Hailey folgt Justin
3 Überprüfe Justin...
4 [3] Beantwortete Anfrage: Justin folgt Selena nicht.
5 Bereits bekannt: Selena folgt Justin
6 [4] Beantwortete Anfrage: Justin folgt Hailey nicht.
7 Bereits bekannt: Hailey folgt Justin
8 Superstar gefunden: Justin

```

superstar2.txt

```
1 [1] Beantwortete Anfrage: Turing folgt Hoare
2 [2] Beantwortete Anfrage: Dijkstra folgt Knuth nicht.
3 [3] Beantwortete Anfrage: Codd folgt Hoare nicht.
4 [4] Beantwortete Anfrage: Dijkstra folgt Codd nicht.
5 Überprüfe Dijkstra...
6 [5] Beantwortete Anfrage: Dijkstra folgt Turing nicht.
7 [6] Beantwortete Anfrage: Turing folgt Dijkstra
8 [7] Beantwortete Anfrage: Dijkstra folgt Hoare nicht.
9 [8] Beantwortete Anfrage: Hoare folgt Dijkstra
10 Bereits bekannt: Dijkstra folgt Knuth nicht
11 [9] Beantwortete Anfrage: Knuth folgt Dijkstra
12 Bereits bekannt: Dijkstra folgt Codd nicht
13 [10] Beantwortete Anfrage: Codd folgt Dijkstra
14 Superstar gefunden: Dijkstra
```

superstar3.txt

```
1 [1] Beantwortete Anfrage: Edsger folgt Jitse
2 [2] Beantwortete Anfrage: Jorrit folgt Peter nicht.
3 [3] Beantwortete Anfrage: Pia folgt Rineke nicht.
4 [4] Beantwortete Anfrage: Rinus folgt Sjoukje nicht.
5 [5] Beantwortete Anfrage: Jitse folgt Jorrit nicht.
6 [6] Beantwortete Anfrage: Pia folgt Rinus nicht.
7 [7] Beantwortete Anfrage: Jitse folgt Pia nicht.
8 Überprüfe Jitse...
9 [8] Beantwortete Anfrage: Jitse folgt Edsger
10 Kein Superstar gefunden.
```

superstar4.txt

```
1 [1] Beantwortete Anfrage: Hanna folgt Melker nicht.
2 [2] Beantwortete Anfrage: Liv folgt Ellen nicht.
3 [3] Beantwortete Anfrage: Ali folgt Lova
4 [4] Beantwortete Anfrage: Vide folgt Freja nicht.
5 [5] Beantwortete Anfrage: Melvin folgt Loke nicht.
6 [6] Beantwortete Anfrage: Sigge folgt Milton
7 [7] Beantwortete Anfrage: Sofia folgt Arvid
8 ...
9 [77] Beantwortete Anfrage: Noel folgt Emilia nicht.
10 [78] Beantwortete Anfrage: Charlie folgt Folke
11 [79] Beantwortete Anfrage: Noel folgt Folke
12 Überprüfe Folke...
13 ...
14 ...
15 ...
16 Bereits bekannt: Charlie folgt Folke
```

17	[227] Beantwortete Anfrage: Folke folgt Penny nicht.
18	[228] Beantwortete Anfrage: Penny folgt Folke
19	[229] Beantwortete Anfrage: Folke folgt Rut nicht.
20	[230] Beantwortete Anfrage: Rut folgt Folke
21	Superstar gefunden: Folke

1.4 Bewertungskriterien

Die Bewertungskriterien (Fettdruck) vom Bewertungsbogen werden hier näher erläutert (Punkt-
abzug in []).

- (1) [-1] **Dokumentation sehr unverständlich bzw. unvollständig.**
- (2) [-1] **Vorgegebenes Dateneingabeformat mangelhaft umgesetzt:**
Das vorgegebene Eingabeformat sollte eingehalten und in eine geeignete interne Darstellung umgesetzt werden. Insbesondere sind Listen dafür geeignet.
- (3) [-1] **Lösungsverfahren fehlerhaft:**
Das Lösungsverfahren, mit dem der Superstar bestimmt wird, sollte korrekt sein. Insbesondere ist vom Verfahren sicher zu stellen, dass der letztlich gefundene Kandidat wirklich alle Superstar-Eigenschaften erfüllt. Die Ermittlung des Superstars bereits während des Einlesens der Daten ist keine Aufgabenlösung.
- (4) [-1] **Strategie zur Verringerung der Anfragenzahl mangelhaft / fehlt:**
Es wurde eine Strategie angewandt, durch die die benötigte Zahl der Anfragen verringert wird. Doppelte Anfragen führen nicht zu Punktabzug, jedoch wird für naive Brute-Force-Strategien mit quadratischem Aufwand (Beziehungenabfragen von jedem zu jedem Kandidaten) ein Punkt abgehoben. Eine Aussage zur Abschätzung der maximal notwendigen Anfragenzahl wird nicht erwartet.
- (5) [-1] **Verfahren bzw. Implementierung unnötig aufwendig / ineffizient:**
Nur besonders umständliche oder ineffiziente Verfahren bzw. Implementierungen führen zu Punktabzug.
- (6) [-1] **Programmausgabe der Superstar-Suche nicht nachvollziehbar:**
Die Ausgabe der Superstar-Suche muss nachvollziehbar sein; eine reine Ja/Nein-Antwort ist zu wenig, aber es genügt, wenn die Anzahl der Abfragen und ggf. der gefundene Superstar ausgegeben werden. Am besten ist, wenn das Programm den gesamten Suchlauf ausgibt. Auch wenn es keinen Superstar gibt, so sollte das Programm dies entsprechend ausgeben.
- (7) [-1] **Beispiele fehlerhaft bzw. zu wenige (mind. 1/3 und 1/1):**
Es wird die Dokumentation der Ergebnisse von mind. 1 der 3 vorgegebenen Beispiele 1, 2 und 4 erwartet sowie das Ergebnis des Beispiels 3 (kein Superstar).