

PseudoCode Software-Praktikum

PseudoCode für PairMatch

Algorithm 1 PairMatch

```
1: procedure ALGORITHMUS UM PAARE ZU ERSTELLEN
2:   Erstelle einen ungerichteten Graphen  $G\{V,E\}$ , mit Teilnehmern als Knoten
3:   for alle Kombinationen an Knoten  $V$  in  $G$  do
4:     überprüfe ob fulfillsHardCriteria=true
5:     if fulfillsHardCriteria=true then
6:       berechne Kosten der jeweiligen Kriterien
7:       Füge Kante zwischen den Solo-Paaren hinzu, mit den Kosten als Gewicht
8:   for alle Knoten  $V$  in  $G$  do
9:     if noch Solos in  $V$  enthalten then
10:      Solo mit geringster Anzahl an Kanten wird gewählt
11:      Finde Kante mit niedrigstem Gewicht welche von diesem Solo ausgeht
12:      Ein Paar Objekt wird aus diesen beiden Solos erstellt und in eine
13:      Liste PairMatched hinzugefügt
14:      Entferne beide Solos aus  $V$  und deren Kante aus  $E$ 
15:   return PairMatched
```

Die Zeilen 2-7 sind für die Erstellung des Graphen verantwortlich. Dabei wird ein Teilnehmender als Knoten repräsentiert und jeweils überprüft, ob für jede Kombination an Teilnehmenden die "HardCriteria" erfüllt werden. Dazu gehört unter anderem, dass ein Paar eine Küche haben muss oder kein Veggie mit einem Fleischesser gematched wird. Werden diese Kriterien erfüllt, werden anhand der Gewichtung die Kosten einer jeweiligen Verbindung berechnet, was in dem Graph als Gewicht einer Kante zwischen den Knoten repräsentiert wird.

Danach wird über die Liste an Knoten iteriert. Als erster Knoten jeder Iteration wird derjenige mit der geringsten Anzahl an Kanten gewählt, danach wird die Kante mit dem niedrigsten Gewicht von diesem Solo aus und der anschließende Knoten betrachtet. Diese beiden Knoten bilden dann ein Paar und werden sie aus der Liste von Solos gelöscht, sowie deren Kante aus der Menge der Kanten.

Wir betrachten jeweils den Knoten mit geringster Anzahl an Kanten, damit sichergestellt wird, dass die maximale Anzahl an matches generiert wird.

PseudoCode für GroupMatch

Algorithm 1 GroupMatch

```
1: procedure ALGORITHMUS UM GRUPPEN ZU ERSTELLEN
2:   Unterteile die Paare in Meat/None und Vegan/Veggie Listen
3:   for Beide Listen do
4:     Erstelle 2 Graphen  $G_1$  und  $G_2$ ,
5:   for alle Kombinationen an Knoten  $V_1$  in  $G_1$  und  $V_2$  in  $G_2$  do
6:     if fulfillsHardCriteria=true then
7:       berechne Kosten der jeweiligen Kriterien
8:       Füge Kante zwischen den "Pärchen-Paaren" hinzu, mit den Kosten als Gewicht
9:   for alle Knoten  $V_1$  in  $G_1$  und  $V_2$  in  $G_2$  do
10:    if noch Pärchen mit 8 Kanten enthalten then
11:      Pärchen mit geringster Anzahl an Kanten wird gewählt (mind. 8 Kanten)
12:      Finde 8 Kanten mit niedrigstem Gewicht welche von diesem Paar ausgehen
13:      Erstelle SuperGroup Objekt mit 9 Pärchen
14:      Aus der SuperGroup werden neun 3er Gruppen erstellt, wobei jedes Pärchen
15:      einmal kocht und bei jedem Gang sich unterschiedliche Pärchen treffen
16:      Entferne die 9 Pärchen aus den jeweiligen Graphen
17:   return GroupMatched
```

Unser GroupMatch verläuft ähnlich wie unser PairMatch. Hier werden jedoch in den Zeilen 2-4 zwei unterschiedliche Graphen erstellt, jeweils mit den Meat/None Paaren und Veggie/Vegan Paaren. Danach wird bei der Erstellung der Kanten und deren Gewichte gleich vorgegangen, wie bei PairMatch.

Bei der Erstellung der "SuperGroups", in denen jeweils 9 Pärchen gespeichert werden wird auch von dem Paar mit den wenigsten Kanten ausgegangen, um die optimale Anzahl an Matches zu generieren. Hier wird jedoch zusätzlich noch überprüft, ob von dem Paar mindestens 8 Kanten ausgehen, da eine SuperGroup aus 9 Paaren bestehen muss.