Генерация полных предпочтений n x n

In[ ]:=
```
Clear@gen
gen[m_] := Module[{distr, mlst = {}, wlst = {}, mProfile = <||>, wProfile = <||>},
  For[i = 1, i ≤ m, i++,
   AppendTo[mlst, "m" <> ToString[i]];
   AppendTo[wlst, "w" <> ToString[i]]];
  For[i = 1, i ≤ m, i++,
   AppendTo[mProfile, mlst[[i]] → <|"Preferences" → RandomSample@wlst, "Quota" → 1|>];
   AppendTo[wProfile, wlst[[i]] → <|"Preferences" → RandomSample@mlst, "Quota" → 1|>]];
  distr = SortBy[Transpose@{RandomSample@mlst, RandomSample@wlst}, First];
  {mProfile, wProfile, distr}]
```

In[ ]:=
```
gen[5]
```

In[ ]:=
```
res = {<|"m1" → <|"Preferences" → {"w5", "w1", "w4", "w2", "w3"}, "Quota" → 1|>,
    "m2" → <|"Preferences" → {"w1", "w5", "w3", "w4", "w2"}, "Quota" → 1|>,
    "m3" → <|"Preferences" → {"w3", "w1", "w5", "w2", "w4"}, "Quota" → 1|>,
    "m4" → <|"Preferences" → {"w5", "w2", "w4", "w3", "w1"}, "Quota" → 1|>,
    "m5" → <|"Preferences" → {"w1", "w4", "w2", "w3", "w5"}, "Quota" → 1|>|>,
   <|"w1" → <|"Preferences" → {"m1", "m4", "m2", "m3", "m5"}, "Quota" → 1|>,
    "w2" → <|"Preferences" → {"m5", "m4", "m3", "m1", "m2"}, "Quota" → 1|>,
    "w3" → <|"Preferences" → {"m2", "m4", "m3", "m1", "m5"}, "Quota" → 1|>,
    "w4" → <|"Preferences" → {"m1", "m5", "m2", "m4", "m3"}, "Quota" → 1|>,
    "w5" → <|"Preferences" → {"m2", "m1", "m3", "m5", "m4"}, "Quota" → 1|>|>,
   {{"m1", "w3"}, {"m2", "w5"}, {"m3", "w1"}, {"m4", "w4"}, {"m5", "w2"}}};
```

Поиск блокирующих пар

In[ ]:=
```
Clear@checkMW
checkMW[mProfile_, wProfile_, distr_] :=
 Module[{lstCheck, lstStep, m, pairs = {}, lstStep2, lstCheck2},
  For[i = 1, i ≤ Length@mProfile, i++,
   m = "m" <> ToString[i];
   lstStep = mProfile[m, "Preferences"];
   lstCheck =
    lstStep[[1 ;; (Position[lstStep, Cases[distr, {m, _}][[1, 2]]][[1, 1]]) - 1]];
   For[j = 1, j ≤ Length@lstCheck, j++,
    lstStep2 = wProfile[lstCheck[[j]], "Preferences"];
    lstCheck2 = lstStep2[[
      1 ;; Position[lstStep2, Cases[distr, {_, lstCheck[[j]]}][[1, 1]]][[1, 1]] - 1]];
    If[Intersection[{m}, lstCheck2] ≠ {}, AppendTo[pairs, {m, lstCheck[[j]]}]]]];
  If[pairs == {}, True, pairs]
 ]
```

In[ ]:=
```
checkMW[res[[1]], res[[2]], res[[3]]]
```

Out[ ]=
```
{{m1, w1}, {m1, w4}, {m2, w1}, {m3, w3}, {m5, w4}}
```

Printed by Wolfram Mathematica Student Edition

```
In[•]:= mProfile = <|
          "m1" → <|"Preferences" → {"w2", "w3", "w1"}, "Quota" → 1|>,
          "m2" → <|"Preferences" → {"w3", "w1", "w2"}, "Quota" → 1|>,
          "m3" → <|"Preferences" → {"w3", "w2", "w1"}, "Quota" → 1|>
          |>;

In[•]:= wProfile = <|
          "w1" → <|"Preferences" → {"m3", "m1", "m2"}, "Quota" → 1|>,
          "w2" → <|"Preferences" → {"m2", "m3", "m1"}, "Quota" → 1|>,
          "w3" → <|"Preferences" → {"m1", "m2", "m3"}, "Quota" → 1|>
          |>;

In[•]:= distr = {{"m1", "w1"}, {"m2", "w2"}, {"m3", "w3"}};
```

Добавление 1 более сильной пары

```
In[•]:= Clear@resOptStep
resOptStep[distr_, pair_] := Module[{a, b, distrNew = distr},
    a = Cases[distr, {pair[[1]], _}][[1, 2]];
    b = Cases[distr, {_, pair[[2]]}][[1, 1]];
    distrNew = DeleteCases[distrNew, {_, a}];
    distrNew = DeleteCases[distrNew, {b, _}];
    AppendTo[distrNew, {b, a}];
    AppendTo[distrNew, pair];
    SortBy[distrNew, First]];
```

Избавление от блокирующих пар

```
In[•]:= Clear@resOpt
resOpt[mProfile_, wProfile_, distr_] := Module[{distrNew = distr, pair, res},
  While[ListQ[checkMW[mProfile, wProfile, distrNew]] == True,
   pair = checkMW[mProfile, wProfile, distrNew][[1]];
   distrNew = resOptStep[distrNew, pair]];
  {checkMW[mProfile, wProfile, distrNew], distrNew}]

In[•]:= resOpt[mProfile, wProfile, distr]
Out[•]= {True, {{m1, w3}, {m2, w2}, {m3, w1}}}

In[•]:= resOpt[res[[1]], res[[2]], res[[3]]]
Out[•]= {True, {{m1, w1}, {m2, w5}, {m3, w3}, {m4, w2}, {m5, w4}}}
```