1a. Egy utazási iroda társasutazására N személy jelentkezik. Vannak azonban olyanok, akik bizonyos okok miatt nem akarnak együtt utazni. Készíts algoritmust, ami megadja a legtöbb embert, akik közül bármelyik kettő hajlandó együtt utazni!

```
T(I,J):=igaz, ha az I-edik ember a J-edikkel akar együtt
utazni, s hamis, ha nem. X(I)=1, ha utazik, 2, ha nem.
Minimumkeresés(N,M,VAN,Y):
  I:=1; X(1..N):=(0,...,0); DB:=0; VAN:=hamis; E:=0
  Ciklus amíg I≥1
    Jó_eset_keresés(M,X,I,melyik,VANJO)
    Ha VANJO akkor  X(I):=melyik; I:=I+1   [előrelép]
           különben X(I):=0;      I:=I-1   [visszalép]
    Ha I>N akkor VAN:=igaz; I:=I-1
                 Ha szumma(X)>E akkor Y:=X; E:=szumma(X)
  Ciklus vége
Eljárás vége.
Jó_eset_keresés(M,X,I,mely,VAN):
  mely:=X(I)+1
  Ha mely=1 és Rossz_eset(I,X) akkor mely:=mely+1
  VAN:=(mely≤2)
Eljárás vége.
Rossz_eset(I,X):
  J:=1
  Ciklus amíg J<I és (X(J)=0 vagy T(I,J))
    J:=J+1
  Ciklus vége
  Rossz_eset:=(J<I)
Eljárás vége.
```

1b. Egy M tagú társaságban tudjuk, hogy kik ismerik egymást. Készíts algoritmust, ami megadja a legtöbb embert úgy, hogy senki ne ismerje a másikat!

```
T(I,J):=igaz, ha az I-edik ember a J-ediket ismeri, s hamis,
ha nem. X(I)=1, ha az I-ediket kiválasztottuk, 2, ha nem.
Minimumkeresés(N,M,VAN,Y):
  I:=1; X(1..N):=(0,...,0); DB:=0; VAN:=hamis; E:=0
  Ciklus amíg I≥1
    Jó_eset_keresés(M,X,I,melyik,VANJO)
    Ha VANJO akkor  X(I):=melyik; I:=I+1   [előrelép]
           különben X(I):=0;      I:=I-1   [visszalép]
    Ha I>N akkor VAN:=igaz; I:=I-1
                 Ha szumma(X)>E akkor Y:=X; E:=szumma(X)
  Ciklus vége
Eljárás vége.
Jó_eset_keresés(M,X,I,mely,VAN):
  mely:=X(I)+1
  Ha mely=1 és Rossz_eset(I,X) akkor mely:=mely+1
  VAN:=(mely≤2)
Eljárás vége.
Rossz_eset(I,X):
  J:=1
  Ciklus amíg J<I és (X(J)=0 vagy nem T(I,J))
    J:=J+1
  Ciklus vége
  Rossz_eset:=(J<I)
Eljárás vége.
```