1. Fie digraful G = (S, E), unde S = multimea indivizilor din societate, iar E = {uv | v ∊ c(u), v si u din S} (u il cunoaste pe v).

Putem rezolva aceasta problema reducand-o la **problema k-colorarii** unui graf, numarul minim de jurii posibil fiind numarul cromatic al grafului G, X(G). Pe noi ne intereseaza numarul maxim de jurii, asa ca vom afla valoarea maxima a indicelui cromatic. Demonstram ca X(G) <= 2k+1

Construim graful G’ = (S, E’), S = multimea indivizilor din societate, iar E’ = {uv | uv ∊ E sau vu ∊ E}; cu alte cuvinte, muchia uv este in E’ daca u il cunoaste pe v, sau v il cunoaste pe u, sau ambii se cunosc intre ei.

Tinand cont de faptul ca in digraful G, orice individ cunoaste maximum k alti indivizi (c(x) <= k, deci ∆(G) = k), in graful G’ avem ∆(G’) = 2 \* ∆(G) = 2k, deoarece gradul maxim al unui nod va fi 2k (el cunoaste maximum k alti indivizi, iar maximum k alti indivizi il cunosc pe el).

Din colorarea unui graf, stim ca X(G’) <= ∆(G’)+1, deci avem X(G’) <= 2k+1. Asadar, numarul maxim de jurii (dat de indicele cromatic al grafului G’) este 2k+1.

O alta idee de rezolvare consta in observarea faptului ca numarul de jurii este egal cu numarul de subgrafuri complete induse in graful complementar al lui G’. Daca un subgraf este complet, inseamna ca nimeni nu cunoaste pe nimeni. In cel mai rau caz, graful complementar lui G’ contine doar noduri izolate => Exista n subgrafuri complete (a cate 1 varf) => n jurii, unde n = |G’|

Stiind ca numarul maxim de jurii este n, iar 2k este maximum n-1 (gradul maxim al unui nod in G’ este n-1, deci 2k e maxim n-1), reiese ca numarul maxim de jurii este 2k+1 = (n-1)+1 = n.

2. a)

b) Dn = (V, E), unde orice x∊V este de forma (x1, x2, …, xn), iar E = {xy | (x2, x3, …, xn) = (y1, y2, …, yn-1).

Fie succ(x) = {y | xy ∊ E}. succ(x) = {(x2, x3, …, xn, 0), (x2, x3, …, xn, 1)} si BFS (x) multimea nodurilor ce pot fi vizitate din x, x de forma (x1, x2, …, xn).

Demonstram ca pentru 2 varfuri aleatorii x, y∊V, exista drum de la x la y in Dn.

BFS (x) = BFS ((x1, x2, …, xn)) =

{(x1, …, xn)} U BFS (succ(x1, x2, …, xn)) =

{(x1, x2, …, xn)} U {(x2, x3, …, xn, 0)} U {(x2, x3, …, xn, 1)} U BFS (succ(x2, …, xn, 0)) U BFS (succ(x2, …, xn, 1)) =

{(x1, x2, …, xn)} U {(x2, x3, …, xn, 0)} U {(x2, x3, …, xn, 1)} U {(x3, …, xn, 0, 0)} U {(x3, …, xn, 0, 1)} U … =

Cum y∊, reiese ca exista drum de la x la y, pentru orice x, y∊V din Dn => Dn tare conex.

Dn admite un parcurs eulerian inchis ⬄ Dn este tare conex si = , pentru orice x∊V

Orice nod x∊V are exact 2 succesori (x2, …, xn, 0), (x2, …, xn, 1) si exact 2 predecesori (0, x1, …, xn-1), (1, x1, …, xn-1) => = = 2.

Cum Dn este tare conex si = => Dn admite un parcurs eulerian inchis

c) naiba stie…

3. G = (V, E)

O observatie cheie ce ne va ajuta in rezolvarea problemei e ca daca doua drumuri P si Q intre (s, t) au aceeasi lungime, atunci diferenta de noduri trebuie sa fie para; |V(P) ∆ V(Q)| = 2\*k, k=>1.

Demonstram ca graful H este conex aratand ca exista drum intre oricare doua varfuri din H, P si Q, prin inductie dupa m, unde m = |V(P) ∆ V(Q)|, m >= 2, unde P si Q reprezinta drumuri (s, t) de lungime d in graful G.

Pentru m = 2 ipoteza este adevarata, intrucat P si Q sunt adiacente in H pentru m = 2.

(\*) Pp. adevarata ipoteza pentru m = 2\*k

Demonstram pentru m = 2\*(k+1)

P este de forma (a1, a2, …, ai, p1, p2, …, pm, a(i+m+1), …, ad)

Q este de forma (a1, a2, …, ai, q1, q2, …, qm, a(i+m+1), …, ad)

Construim R, astfel incat |V(R) ∆ V(Q)| = 2 si |V(P) ∆ V(R)| = 2\*k

R este de forma (a1, a2, …, ai, q1, …, qj, p(j+1), p(j+2), q(j+3), …, qm, a(i+m+1), …, ad)

Stim sigur din modul in care P este format ca exista muchie intre p(j+1) si p(j+2). Deoarece qj si pj sunt la aceeasi distanta fata de s, oricare j din [1, m], si datorita faptului ca G este **cordal** rezulta ca avem o coarda intre qj si p(j+1), iar o alta coarda intre p(j+2) si q(j+3). Daca acest lucru nu s-ar intampla, atunci ar exista un C4 fara o coarda.

Avem asadar d(R) = d(Q) = d(P) = d

Se poate observa din forma lui R ca |V(R) ∆ V(Q)| = 2 si |V(P) ∆ V(R)| = 2\*k. Din ipoteza (\*) presupusa adevarata, stim ca exista un drum intre P si R (m = 2\*k). Din enunt stim ca R si Q sunt adiacente, deoarece |V(R) ∆ V(Q)| = 2. Rezulta asadar ca exista drum intre P si R.

In concluzie, avem drum intre oricare doua drumuri P si Q, |V(P) ∆ V(Q)| = 2\*k => H graf conex

4. Pentru o zi k ∊ {1, 2, 3, 4}, valoarea maxima a lui color(v) este n (initial 5000). Pozitia i pe care se gaseste un bit diferit este cel mult [log2(n)], deci pana acum, noua valoarea a lui color(v) = [log2(n)]. Adaugand si bitul care difera, color(v) poate fi maxim 2 \* [log2(n)] (parte superioara).

Putem deduce astfel urmatoarele:

Initial, color(v) ∊ {1, 2, …, 5000}, ⩝ angajat v

k = 1 => color(v) ∊ {0, 1, …, 26}, ⩝ angajat v

k = 2 => color(v) ∊ {0, 1, …, 10}, ⩝ angajat v

k = 3 => color(v) ∊ {0, 1, …, 8}, ⩝ angajat v

k = 4 => color(v) ∊ {0, 1, …, 6}, ⩝ angajat v

Dupa orice zi k, pentru ⩝ angajat v!=v0, noua culoare a lui v va fi diferita de cea a lui boss(v); color(v) != color(boss(v)).

Pp. prin reducere la absurd ca asta s-ar intampla. Asta inseamna ca, dupa transformare, reprezentarea binara a noilor valori color(v) si color(boss(v)) sunt identice.

Inseamna ca pozitia i pentru valorile initiale color(v) si color(boss(v)) este identica cu pozitia i pentru valorile initiale color(boss(v)) si color(boss(boss(v))), iar bitul de pe pozitia i din reprezentarea in baza 2 a lui color(v) initial = bitul de pe pozitia i din reprezentarea in baza 2 a lui color(boss(v)) initial.

Dar dupa modul in care construim noua culoare color(v), stim ca bitul de pe pozitia i din color(v) initial != bitul de pe pozitia I din color(boss(v)) initial. Avem, deci, contradictie.

Analizam pentru urmatoarele 3 zile (5, 6, 7) valorile pentru color(v) in a k-a zi (dintre acestea).

color(v) = color(boss(v)). Momentan, singurele 2 noduri care au aceeasi culoare sunt v0, si subalternii directi ai lui v0

v0 isi alege o culoare din {0, 1, 2}

k=1 => ⩝ angajat v cu color(v) = 5 va avea acum color(v) ∊ {0, 1, 2} != color(boss(v))

k=2 => ⩝ angajat v cu color(v) = 4 va avea acum color(v) ∊ {0, 1, 2} != color(boss(v))

k=3 => ⩝ angajat v cu color(v) = 3 va avea acum color(v) ∊ {0, 1, 2} != color(boss(v))

Orice culoare noua pe care v si-o alege difera de cea trimisa subalternilor lui

In a opta zi, avem color(v) ∊ {0, 1, 2}, ⩝ angajat v. Din modul in care fiecare v si-a ales culoarea, stim ca color(v) != color(boss(v)), ⩝ angajat v!=v0.

Construing arborele G = (V, E), V = multimea angajatilor, |V| = 5000, iar E = {uv | u = boss(v)}, pentru orice varf v avem color(v) ∊ {0, 1, 2} si color(v) != color(boss(v)), ⩝ v!=v0. Asadar, protocolul este o 3-colorare.