1. Fie graful orientat G = (V, E) creat astfel: fiecare individ are asociat un nod, iar E este formata din toate perechile (a, b) cu propritatea ca individul a cunoaste pe individul b. Din problema deducem ca fiecare nod are gradul de iesire cel mult k.

Demonstram urmatoarea lema: exista un nod in G cu gradul maxim 2k.

Intr-adevar, in caz contrar, fiecare nod are gradul ≥2k+1, deci suma gradelor ar fi ≥n(2k+1)

Din fiecare nod pleaca cel mult k muchii, deci G contine maxim nk muchii. Deoarece suma gradelor nodurilor este egala cu dublul numarului de muchii, obtinem inegalitatea 2nk ≥ n(2k+1) ⬄ n≤0, contradictie.

Aratam prin inductie dupa n = |V| ca exista o 2k+1 colorare a grafului G.

Pentru n = 1, coloram singurul nod cu culoarea 1. Deoarece 2k+1≥1, aceasta este o 2k+1 colorare.

Presupunem ca pentru orice graf cu cel mult n noduri exista o 2k+1 colorare si aratam pentru un graf H, cu n+1 noduri.

Fie v un nod cu gradul cel mult 2k (folosim lema). Coloram cu 2k+1 culori graful H' obtinut din H prin eliminarea lui v (pasul de inductie). Deoarece v are cel mult 2k vecini, acesta poate fi colorat cu una din cele 2k+1 culori, deci am obtinut o 2k+1 colorare pentru H.

Pentru fiecare culoare din cele maxim 2k+1, formam un juriu cu toate nodurile de respectiva culoare. Se observa ca nu pot exista 2 indivizi intr-un juriu astfel incat unul sa il cunoasca pe celalalt si fiecare individ face parte din exact un juriu, deci aceasta este o solutie a problemei.

2. a) Atasat la foile printate

b) = (V, E), unde ⩝ x∊V este de forma (, , …, ), iar E = {xy | (, , …, )=(, , …, ).

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

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

BFS (x) = BFS (, , …, ) =

{(, , …, )} U BFS (succ(, , …, )) =

{(, , …, )} U {(, , …, , 0)} U {(, , …, , 1)} U BFS (succ(, , …, , 0)) U BFS (succ(, , …, , 1)) =

{(, , …, )} U {(, , …, , 0)} U (, , …, , 1)} U {(, …, , 0, 0)} U {(, …, , 0, 1)} U … =

Cu alte cuvinte, prin n shiftari la stanga (n BFS-uri), putem obtine orice secventa de 0 si 1.

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

admite un parcurs eulerian inchis ⬄ este tare conex si = , pt. ⩝ x∊V

⩝ nod x∊V are exact 2 succesori (, , …, , 0), (, , …, , 1) si exact 2 predecesori (0, , , …, ), (1, , , …, )}=> = = 2.

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

c) Observam ca un nod din este determinat unic de o secventa de n valori de 0 si 1. Avem deci 2^n noduri in total pentru . Pentru fiecare varf v, avem exact 2 arce care il au pe v ca extremitate initiala. Avem astfel 2^n\*2 = 2^(n+1) arce distincte in total

L() si au, asadar, acelasi numar de varfuri, 2^(n+1).

Pentru orice arc din care pleaca din varful x si adauga la sfarsit bitul b, ii asociem varful (, , …, , b) in L(. Pentru fiecare varf x din , vom avea 2 varfuri corespondente in L(): (, , …, , 0) si (, , …, , 1).

Exista o functie bijectiva f : V(L()) -> V() care pentru ⩝ varf y din de forma (, , …, ) asigneaza un corespondent unic x in L() de forma (, , …, , b), a.i. =, =, …, =, =b.

f(, , …, ) = (, , …, , b)

Exista o functie g : E(L()) -> E() care pentru fiecare arc ((, , …, , b), (b , …, )) din L() asigneaza

g ((, , …, , b), (b, , …, ))= ( f(, , …, , b), f(b, , …, )).

Avem, deci, L() izomorf cu

Stim ca este eulerian si ca line-graph-ul oricarui graf eulerian e hamiltonian. Deducem de aici ca L() admite un ciclu hamiltonian.

3. Aratam ca daca un nod v arbitrar apare pe pozitia i intr-un (s, t) path de lungime d, atunci el nu poate aparea pe o pozitie j ≠ i intr-un alt (s, t) path de lungime d.

Daca prin absurd ar fi doua (s, t) path-uri de lungime d

P = (s = , , ..., v = , ..., = t)

P' = (s = , , ..., v = , ..., = t) (v si sunt adiacente)

i < j

atunci Q = (s = , , ..., v = , , ..., = t) ar fi un (s, t) path de lungime d-(j-i) < d (contradictie)

Fie doua (s, t) path-uri P si P' arbitrare de lungime d. Aratam ca exista drum intre ele prin inductie dupa numarul de pozitii pe care difera (jumatate din diferenta simetrica).

Fie i prima pozitie pe care difera (i > 1) si j prima pozitie dupa i a.i. drumurile sunt egale incepand cu pozitia j incolo (i < j < d)

P = (s = , , ..., , , , ..., , , , ..., = t)

P’ = (s = , , ..., , , , ..., , , , ..., = t)

Ciclul (, , ..., , , ..., ) contine doar noduri distincte si are lungime (j - i + 1 + j - i + 1) >= 4, deci contine cel putin o coarda, deoarece graful e cordal.

Lg ciclu >=4

Daca lungimea ciclului = 4, inseamna ca |V(P) ∆ V(P’)| = 2, deci avem muchie directa de la P la P’, conform enuntului.

Pp. ca exista drum intre oricare 2 (s, t) path-uri care difera in k varfuri (j-i = k)

Demonstram ca exista drum intre P si P’ care difera in k+1 varfuri.

G fiind graf cordal, stim sigur ca pentru oricare 2 noduri pe pozitii consecutive din P si P’ exista 1 coarda care le leaga: (, ) sau (, ). Daca aceasta coarda leaga 2 varfuri egale ( = sau = ), atunci afirmatia este adevarata intrucat avem muchiile (, ) si (, )

Daca ar exista o coarda intre 2 varfuri la distanta >=2, atunci ar exista un (s, t) path cu (s, t) < d (contradictie)

Deci, exista un drum Q de forma

P = (s = , , ..., , , , ..., , , , ..., = t)

P’ = (s = , , ..., , , , ..., , , , ..., = t)

Q = (s = , , ..., , , , …, , , , …, = t)

Avem Q si P’ difera intr-un varf, iar Q si P difera in k varfuri. Q si P’ sunt adiacente din enunt, iar intre Q si P exista drum prin ipoteza presupusa adevarata => exista drum intre P si P’.

Avem drum pentru oricare 2 drumuri P si P’ => H conex

Sa presupunem ca diametrul lui H >= d. Asta inseamna ca exista doua drumuri P si P’ care difera in cel putin d varfuri. Stim ca orice drum este format din d+1 noduri, primul fiind s si ultimul fiind t. Raman astfel d+1-2 = d-1 varfuri in care cele doua drumuri pot diferi => contradictie => doua drumuri P si P’ pot diferi in maxim d-1 noduri.

Conform inductiei de mai sus, exista intre P si P’ un drum de lungime d-1 => lungimea maxima intre oricare 2 drumuri = d-1 => diametrul lui H = d-1.

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)] + 1.

Putem deduce astfel urmatoarele:

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

k = 1 => color(v) ∊ {0, 1, …, 25}, ⩝ angajat v

k = 2 => color(v) ∊ {0, 1, …, 9}, ⩝ angajat v

k = 3 => color(v) ∊ {0, 1, …, 7}, ⩝ angajat v

k = 4 => color(v) ∊ {0, 1, …, 5}, ⩝ 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. De asemenea, max(color(v)) = 5

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)), iar max(color(v)) = 4

k=2 => ⩝ angajat v cu color(v) = 4 va avea acum color(v) ∊ {0, 1, 2} ≠ color(boss(v)), iar max(color(v)) = 3

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

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.