ALGORITMOEN DISEINUA

G5: PROGRAMAZIO DINAMIKOAREN TEKNIKA

1) *f(1)=1* izanik, zein programazio teknika erabiliko zenuke $f(n) = n + \frac{2'}{n} \sum_{i=1}^{n-1} f(i)$ kalkulatuko lukeen programa bat idazteko? Programa horren exekuzio ordena zein izango litzatekeen laburki arrazoitu.

2) Demagun *Partiketa* funtzioa honela definituta dagoela:

```
type Bektore is array (Integer range <>) of Positive;
function Partiketa (B: Bektore ) return Integer is
begin
  for I in 1.. N loop
    if Batu(J=1,I-1,B(J))=Batu(J=I,N,B(J)) hen return I;
    end if;
  end loop;
  return 0;
end Partiketa;
```

Zenbaki osoko positiboen B bektorea emanik, Partiketa funtzioak ondorengo baldintza betetzen duen I (B bektoreko) posizioa itzultzen du:

I posizioaren ezkerraldean dauden B-ko osagaien batura (B(I) balioa kanpo) eta
I posizioaren eskuinaldean dauden B-ko osagaien (B(I) balioa barne)
baturak berdinak dira.

Aurreko baldintza betetzen duen posiziorik ez badago bektorean, funtzioak 0 itzuli beharko du.

a) Frogatu azaldutako ezaugarriak betetzen dituen posizio bakar bat existi daitekeela gehienez.

b) Programaren exekuzio-denbora eta denbora-ordena kalkula bitez.

c) O(*n*) ordena hartuko duen algoritmo bat idatz bedi

d) O(*lg n*) denbora hartuko lukeen algoritmorik aurkitzea posible ote litzatekeen laburki eztabaidatu.

3) $m \times n$ tamainako labirinto batean gora, behera, eskuinera eta ezkerrera mugimenduak bete ditzake errobota batek baldin eta noranzko horretan ormarik ez badago. Labirinto konkretu bat emanik, irteera posizioa eta helmuga posizioak emanik, zein agindu

sekuentziak bete behar dituen errobotak kalkulatuko duen algoritmoa idatzi backtrack eskema erabili gabe.

4) *X* eta *Y* zenbaki positiboak izanik, demagun ondorengo programa:

```
function Nagusia (Znbk: positive) return real is
  Taula: array (1..Znbk) of real:=(1=>X;2=>Y; others=>0);
  function LAG (Znbk: natural) return real is
  begin
    if Taula(Znbk-1)=0 then Taula(Znbk-1):= LAG(Znbk-1); end if;
    if Taula(Znbk-2)=0 then Taula(Znbk-2):= LAG(Znbk-2); end if;
    Taula(Znbk):= (Taula(Znbk-1)+Taula(Znbk-2))/2;
  end LAG;
begin
  if Znbk≤2 then return Taula(Znbk);
  else return LAG(Znbk); end if;
end Nagusia;
```

a) *Nagusia(n)*-k kalkulatzen duen funtzioa definitu.

b) Aurreko programaren diseinurako, zein teknika erabili da?

c) *Nagusia(n)*-ren egikaritzapen-denbora ordena kalkula ezazu.

d) Sarrerako datuak *X=1* eta *Y=1* balira, aurreko programa erabiliko zenuke? Zergatik?

5) (a) $b_1, b_2, ..., b_N$ *N* txanponen klaseak emanik eta jakinik txanpona klase bakoitzeko behar hainbat txapona eskuragarri ditugula, *K* kopuru finkoa emango duten *txanpon kopuru* minimo bat kalkulatuko duen algoritmo bat idaztea eskatzen da. Algoritmoaren *T(N)* kalkulatu. (*K* konstantea dela suposatuko da.).

(Aholkua: izan bedi *Z(K)* *K* batzen duten txanpon zenbakirik txikiena. Aztertu bedi zer erlazioa dagoen *Z(K)* eta $Z(K-b_i)$-ren artean, non $1 \le i \le n$ eta $b_i = K$).

(b) Eta *i* klase bakoitzeko txanpona kopurua m_i balioaz mugaturik balego? Azter ezazu egoera berria eta muga horiek kontuan hartzen dituen ekuazio sistema ere defini ezazu.

6) (a) $b_1, b_2, ..., b_N$ *N* txanponen klaseak emanik eta jakinik txanpona klase bakoitzeko behar hainbat txapona eskuragarri ditugula, *K* kopuru jakin bat itzultzeko zenbat txanpon *konbinazio* desberdin dauden kalkulatuko duen algoritmo bat idaztea eskatzen da.

Adb.: Txaponen klaseak honakoak balira: b_1= 25, b_2= 50, b_3= 75 eta b_4= 100

Balioa	Txanpon konbinazio desberdinak
100	5 modu: 25 + 75, 2 • 50, 100, 4 • 25, 50 + 2 • 25
116	0, ez dago konbinaziorik

Iradokizuna: 25+75 konbinazioa 75+25 konbinazioaren berdina da, eta ondorioz, konbinazio bakarra kontsideratu behar da.

(b) Eta *i* klase bakoitzeko txanpona kopurua m_i balioaz mugaturik balego? Azter ezazu egoera berria eta muga horiek kontuan hartzen dituen ekuazio sistema ere defini ezazu.

7) *N* objektuk p_1, p_2, ..., p_n pisuak eta b_1, b_2, ..., b_n balioak erlazionaturik dituzte hurrenez hurren. Bestalde, *E* edukiera duen motxila bat ere badugu. Helburua, motxilaren edukiera gaindituko ez duten objektuen azpimultzo bat bilatzea da haien balioen batura maximizatuz. (Backtracking eskema erabiltzen dituzten soluzioak ez ditugu onartuko.)

8) $\{b_1,\ldots,b_n\}$ balioak dituzten *n* zigilu mota ditugu. *Z* posta zenbatekoa duen eskutitza, zenbat modu desberdinetan frankea dezakegu? (Postaren frankeoa zehazki *Z* izan behar du, eta noski, zigiluen itsaspen ordenak ez du garrantzirik).

   a) Programazio dinamikoa teknika erabiliz gara ezazu soluzio bat eta haren denbora ordena eta memoria espazio estra kalkula itzazu.

   b) Gainera, defini dituzun indukzio ekuazioak erabiliz eta honako datuokin (b_1=2, b_2=3, b_3=5 eta b_4=8 zigilu motak eta *Z*=21 posta zenbatekoa) dagozkion balioz osatu memoria metatze egitura (iteratiboki edo errekurtsiboki).

9) Unibertsitate bateko sail batek *IK* ikastordu eman behar ditu. Horretarako, *N* klaseko desberdinetako irakasleak kontrata ditzake. *I* klaseko irakasle bakoitzak H_i ordu irakatsi ditzake gehienez lan horregatik beti P_i ogerleko eskuratuz. *IK* klaseak emateko eta ordaindu beharreko ogerleko kopurua minimoa izan dadin, zenbat irakasle kontratatu beharko liratekeen kalkulatuko duen algoritmo bat idatzi. Aldi berean, *N* irakasle klase existitzen direla jakinda, algoritmoaren exekuzio-denboraren ordena kalkulatu.

   Aurreko algoritmoa egoki ezazu (eta berridatzi), gainera, klase bakoitzeko zenbat irakasle kontratatu behar diren jakiteko.

10) Lagunak lanean enpresako zuzendariak Marigorriri aholkularitza bat egin dezala eskatu diola eta, honi, enpresa afari bat antolatzea bururatu zaio. Gonbidatuen zerrenda egiteko, gainbegirale funtzio bat erabil dezake. Enpresako langileriaren egitura hierarkikoa denez, funtzio honen heina piramide bat osatzen du haren gailurrean gaur egungo zuzendaria egonik. Bestalde, administrazioko langileek, langile guztiek erlazionaturik duten jendetasun balioarekiko, lantegiko langile guztiak ordenatu dituzte zerrenda bat osatuz. (Jendetasun balioa zenbaki erreala da.) Afarira joango diren partaideak, ekintza hura goza dezaten, zuzendariak langile bat eta honen gainbegirale zuzena, biak, afarian egon ez daitezela murriztapena ezarri du:

   a) Zerrenda kalkulatuko duen algoritmoa idatz bedi. Helburua, partaideen jendetasun balioa maximizatzea izango da. Exekuzio denbora orden kalkula bedi.

b) Marigorrik nola ziurta dezake partaideen artean enpresako zuzendaria egongo dela?

11) *K* hasierako kopuru bat emanik eta honi *×2, ×3, ×5* eragiketak behin eta berriro aplikatuz, zer balio *X* lor dezakegu *M* baliotik hurbilen dagoena baina hau gainditzen ez duena? Programazio dinamikoa teknika erabil ezazu *X*-ren balioa mugatzeko eta proposatutako algoritmoaren denbora-ordena kalkulatu.

   Aurreko algoritmoa egoki ezazu beharrezkoak diren aginduak gehituz, gainera, Z balio maximo hori ematen duten biderkagaiak zeintzuk diren jakiteko.

12) Demagun *K* ogerleko ditugula eta *n* produktu eskaintzen dizkigutela non p_1,..., p_n hauen prezioak diren. Programazio dinamikoaren teknika jarraituz diseinatu algoritmo bat azaldutako *K* kopuruarekin zenbat produktuen *erosketa konbinazio desberdin* egin ditzakegun mugatzeko.

   Adibidez:

   K=15 eta [p_1= 10, p_2= 7, p_3= 3, p_4=5] kopurua eta produktuak edukiz hurrenez hurren, ondorengoek 3 erosketa desberdin adierazten dituzte:

```
1) [p1= False, p2= False, p3= False, p4= False]
2) [p1= False, p2= True, p3= True, p4= True]
3) [p1= True, p2= False, p3= False, p4= False]
```

   Zure algoritmoak zer erantzungo luke adibide honentzat.

13) Izan bedi *T N* zenbaki osoko desberdinez osatutako taula bat. *T*-ko osagaiak (eta bertan agertzen diren ordena mantenduz) ordena gorakorra jarraituz ordenaturik dauden sekuentzia desberdinen artetik sekuentzia luzeena kalkulatzea eskatzen da. Adibidez, sekuentzia 11,17,5,8,6,4,7,12,3 denean, algoritmoak 5,6,7,12 sekuentzia itzuliko du. Algoritmoaren denbora-ordena zenbatekoa da?

14) L_1, ..., L_n luzera osokoak dituzten *n* barra ditugu. *K* luzerako barra bat behar dugu. Barrak ezin dira moztu baina bai soldatu haien ertzetatik. *K* luzera duen barra eraikitzeko gure barren azpimultzorik existitzen duen ala ez mugatuko duen soluzioa garatu programazio dinamikoaren teknika erabiliz. Algoritmoaren exekuzio ordena eta erabili- tako memoria espazio estra.

   OHARRA: Guztiz garrantzitsua da programazio dinamikoaren teknikaren pausoz pausoko bilakaera zuzena egitea.

15) S-ko azpisekuentzia bat S-ko edozein posizioetatik edozein osagai kopuru kenduz lortzen da. Adib.: *bbacabxxb* sekuentziako azpisekuentzia bat *acxb* da. Bi sekuentzia

emanik hauen azpisekuentzia komun luzeena mugatzen du ondorengo f funtzio errekurtsiboaren definizioak (kasuka):

$f(R,\varepsilon)=\varepsilon$

$f(\varepsilon,S)=\varepsilon$

$f(a\bullet R,a\bullet S)= a\bullet f(R,S)$

$f(a\bullet R,b\bullet S)=f(R,b\bullet S)$ *eta* $f(a\bullet R,S)$ azpisekuentzien artetik luzeena $(a = b)$

Adib.: *f(belarri, berdela) = bela*, *f(kamamila, motxila) = mila*, *f(bai, ez) = ε*.

(Iradokizuna: Luzera luzeeneko azpisekuentzia bat baino gehiago existitzen badu, azpisekuentzia bakarra kalkulatzen du.)

f(R,S) sekuentziaren *Luzera* kalkulatuko duen eta programazio dinamikoa teknika jarraituz diseinaturiko algoritmoa idaztea eskatzen da (memoriadun funtzioak erabili gabe). Zein da algoritmoaren exekuzio-denbora ordena? Eta erabilitako memoria espazio estraren ordena?

16) Izan bedi *G* grafo zuzendua: *G=(Erp, Ark)*. *G* grafoaren adierazpena Auzokideen ma trize baten bidez ematen da. Problema *Dij* matrize logikoa definitzea da, non *Dij* egiazkoa izango den *i*-tik *j*-ra gutxienez bideren bat existitzen bada eta faltsua bestela. Egoki bedi *Floyd*-en algoritmoa problema hau ebatz dezan. Bere exekuzio-denbora azter bedi.

17) *G=(Erp, Ark)* grafo zuzenduaren arkuen pisuak negatiboak ez direla dakigu eta $|Erp|= n$. G grafoa L auzokidetza matrize bidez emanik dator. i, j ∈ Erp multzoko erpin bikote bakoitzaren arteko distantzia minimoa kalkulatu nahi dugu honako formula erabiliz:

D_{ij}^{p}: *i*-tik *j*-rainoko distantzia gehienez *p* pauso eman ditzakegunean.

Oinarrizko balioak $D_{ij}^{p}=L_{ij}$ dira eta *i* eta *j* guztientzat D_{ij}^{n-1}balioak bilatu nahi ditugu, jakinez D_{ij}^{p} errekurtsiboki honela definitzen dela.

Problema programazio dinamikoa teknika erabiliz ebaztea eskatzen da (memoriadun funtzioak erabili gabe), non goian azaldutako ekuazio errekurtsiboa jaso beharko duen; hots, ondorengo puntu guztiak eskatzen dira beraz:

a) tarteko emaitza zein datu-egituratan metatuko den mugatzea: bektorea ala matrizea behar den, egituraren posizio bakoitzak zer adierazten edo metatzen duen, egituraren haizeratzea, nola betetzen joango den, soluzioa non dagoen.

b) emandako definizio errekurtsiboa jarraituz (a)-n aukeratutako datu-egitura betetzen duela ikusi eta soluzioa itzultzen duen algoritmo iteratiboa eman.

c) algoritmoaren denbora eta espazio estra ordenak kalkulatu: O(T(*n*)) eta O(MEE(*n*)).

18) Nilo ibaian *N* kai daude, bakoitzean Y ontzi aloka daitezkeelarik ibaian behera dauden beste kaietara joateko (ibaian gora ezin da joan, korronte handia du eta ibaiak). Edozein A abiapuntu-kaitik ibaian behera dagoen edozein H kaira iristeko ontzi bakar bat alokatzea garestiagoa gerta liteke Atik A+1ra, A+1etik A+6ra eta A+ 6tik H-ra joatea baino, adibidez.

A abiapuntu-kai guztietatik H helmuga-kai guztietara joateko bidai kostu posible guztien artetik bidai kostu minimoak kalkulatzea eskatzen da, bai funtzio memoriadunak erabiliz, bai hauek erabili gabe, baina bi kasuetan programazio dinamikoa teknika erabiliz. Ondoren, exekuzio-denboraren eta memoria-espazio estraren ordenak kalkula bitez.

19) Arkuen pisuak positiboak dituen grafo zuzendu azikliko bat emanik, *Floyd*-en algoritmoak erpin bikote guztientzat bikoteko erpinen arteko distantzia minimoak kalkulatzen ditu. *Floyd*-en algoritmoa eguneratzea eskatzen da, egiten duenaz gain, distantzia minimoa duten bide kopurua kalkula dezan.

20) *N* biteko eta jarraian bi *1*eko bitak ez dituzten kate guztiak eraiki nahi ditugu. *Programazio dinamikoa*ren teknika erabil ezazu kalkulatzeko eraiki genitzakeen horrelako kate desberdinen kopurua baina kateak eraiki gabe. Algoritmoaren denbora-ordena eta erabilitako memoria espazio estra kalkula itzazu.

21) A alfabetoa A={a,b,c} izanik, ondorengo taula, alfabetoko bi karaktereen konbinazioak zertara berridatz daitezkeen jasotzen du:

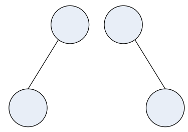
aa »» b	ab »» b	ac »» a
ba »» c	bb »» b	bc »» a
ca »» a	cb »» c	cc »» c

Berridazketa eragiketak, nahiz alfabetoan itxian izan, ez ditu trukaketa eta elkarketa propietateak betetzen.

$X=x_1...x_n$ karakterez osaturiko karaktere katean parentesiak zein posiziotan idatzi beharko liratekeen katea a-ra berridazteko mugatuko du idatzi behar den algoritmoak. Horrela, X= *bbbba* bada, algoritmoak bai itzuli beharko luke eta, adibidez, *(b(bb))(ba)* »» a (berridazketa ordena bat baino gehiago existi baitaiteke: *(b(b(b(ba))))* »» *a*. *N* berridatzi behar den karaktere katearen luzera bada, algoritmoaren denbora-ordena zein da?

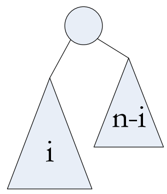
22) $A_1,A_2,...,A_n$ matrize segida biderkatzean, biderkatze eskalar kopurua minimizatzen duen algoritmoa gogora ezazu. Algoritmo horrek *Faktor(1..n, 1..n)* taula eraikitzen du, non *Faktor(i, j)* = *k* baldin *i*-garren eta *j*-garrenaren artean dauden matrizeen faktorketa optimoa $(A_i...A_k)(A_{k+1}...A_j)$ bada. *Faktor(1..n,1..n)* taula emanik $A_1,A_2,...,A_n$ biderkaduraren parentesiak jartzeko era optimoa inprimatzen duen algoritmoa idatz eta azter ezazu.

23) **N** adabegi dituzten zenbait zuhaitz bitar desberdin osa daitezke? Izan bedi Z(n) kopuru hori. Badakigu Z(0)=1 dela -zuhaitz hutsa-, Z(1)=1 -erroa soilik duen zuhaitza-, eta Z(2)=2 dela:



Z(3) eta Z(4) kalkulatzeko zuhaitzak marraz ditzakezu:

a) Z(n) kalkulatzen duen definizio errekurtsiboa idatzi ondorengo diagrama kontuan edukiz:



Orain errekurtsio-ekuazioa zuzena dela n=3, 4 balioentzat ziurta dezakezu zure marrazkiekin.

b) Programazio dinamikoa teknika erabiliz eta zuk aurreko atalean definitutako errekurtsioa jarraituz Z(n) kalkulatzen duen algoritmoa idatzi. Soluzioaren konplexutasuna kalkulatu.

24) Ariketa honetarako *Bikotetar* zuhaitzak erabiliko ditugu, non zuhaitz bat *Bikotetarra* den baldin eta zuhaitzeko barne adabegi orok zehazki bi ume baditu. Hori horrela, N hosto dituzten zenbat *Bikotetar* zuhaitz eratu litezke?

Programazio dinamikoaren teknika erabiliz zenbaki hori kalkulatuko duen algoritmoa egizu. Algoritmoaren denbora-ordena eta erabilitako memoria espazio estra kalkula itzazu.

25) $[h_i,b_i)$ $1\leq i\leq n$ ekintzak izanik, ekintza-hautaketa problema programazio dinamikoa teknika bidez ebaztea eskatzen da. Algoritmoa $Zenb_i$ i=1,2,..,n zenbakiaren kalkulu iteratiboan oinarrituko da, non $Zenb_i$ zenbaki horrek *1,2,..,i* ekintzen multzoko kardinalitate handieneko eta elkar bateragarri diren ekintzen kopurua metatuko duen. Ekintzak bukaera denbora balioekiko ordena gorakorra jarraituz ordenatuta daudela suposatu: $b_1\leq b_2\leq...\leq b_n$.

Soluzio honen eta problema beraren teknika jale bidezko soluzioaren (hain zuzen, lehenen amaitzen den ekintza eta aldez aurretik onartutako ekintzekin teilakatzen ez den hautaketa funtzioa duen soluzio jalea) exekuzio-denbora eta exekuzio-ordena konpara bitez.

26) Al'Telmoko zinemaldi maratoia hurbil dagoela eta, honek irauten dituen hiru egunetan zehar zeintzuk filma ikusiko ditugun planifikatu behar dugu kontu handiz, aldez aurretik sarrerak erosi ahal izateko. i pelikula bakoitza ($1 \leq i \leq n$) behin soilik proiektatuko da, HT_i unean hasiko delarik eta BT_i bukatuz. Gure iritzi-emaile gogokoenak pelikula bakoitzari Si izar esleitu dizkio, eta gure helburua ikusiko ditugun pelikulen izarren kopurua maximizatzea da.

Programazio dinamikoaren teknika aplika ezazu aipatutako helburua betetzen duen programazio bat egiteko, jakinik bi pelikulen artean ordubeteko tartea utzi behar dugula atsedena hartzeko.

Sinplifikatzeko ordu guztiak zinemaldia hasten den eguneko 0:00 orduarekiko neurtuko dira; hots, 0tik 72ra. Gainera, orduak osoak edo erdiak izango dira soilik, hau da, *i* pelikularen hasiera ordua HT_i=34,5 balitz, bigarren eguneko goizeko 10:30etan hasiko dela adieraziko luke.

27) Lantegi bateko gerenteak egun konkretu batean zeintzuk produktu laboratzea komeni den planifikatu behar du. Enpresako datu-basean honako informazioa dago:

- *T[1..n]* taula, non *T[i]*-k *i* klaseko ($1 \leq i \leq n$) produktu bat laboratzeko behar diren minutuak jasotzen duen,
- *On[1..n]* taula, non *On[i]*-k *i* klaseko ($1 \leq i \leq n$) produktu baten fabrikazioak ematen duen onura jasotzen duen, eta
- *Os[1..n]* taula, non *Os[i]*-k egun horretan *i* klaseko ($1 \leq i \leq n$) gehienez labora litezkeen produktu kopurua adierazten duen, lantegiko oinarrizko osagaien izakinak kontuan izanik.

Egun horretan eta planifikazio egoki baten ondorioz lantegiak lor dezakeen onura maximoa kalkulatuko duen algoritmo bat diseina ezazu programazio dinamikoaren teknika erabiliz.

Onura maximoaz gain, aurreko algoritmoa egoki ezazu kalkula dezan zeintzuk produktu komeni diren fabrikatzea onura hori lortzeko bai eta klase bakoitzeko zenbat produktu ere.

28) Garraio enpresa batek *M* kaxaz osatutako bidalketa garrantzitsu bat egin behar du. Bidalketa egiteko *N* garraio gailu mota du. i garraio motak K_i kaxa garraia ditzake gehienez. i garraio gailu baten desplazamenduaren kostua finkoa da, nahiz ia hutsik nahiz topetaraino beterik joan, P_i izanik kostu hori. *M* kaxak kostu minimoz garraiatuko duen garraio konbinazioa kalkula ezazu algoritmo batez eta soluzioaren denbora eta memoria kostuak aztertu.

29) Ariketa honetarako Bikotetar zuhaitzak erabiliko ditugu, non zuhaitz bat Bikotetarra den baldin eta zuhaitzeko barne adabegi orok zehazki bi ume baditu. Hori horrela, N hosto dituzten zenbat Bikotetar zuhaitz eratu litezke?

Programazio dinamikoaren teknika erabiliz zenbaki hori kalkulatuko duen algoritmoa egizu. Algoritmoaren denbora-ordena eta erabilitako memoria espazio estra kalkula itzazu.

30) Inbertsio enpresa batek fondo bat du M hamarreko eurokoa P produktu finantzario artean inbertitzeko. Produktu bakoitzak errentagarritasun espezifikoa du hartan inbertitutako diruaren arabera aldakorra dena. Enpresak matrize bat eraiki du *Errenta*(1..*M*,1..*P*), non *Errenta*(*k*,*f*) urteko mozkinaren portzentaia den, *k* hamarreko euroko ezarpena egiten denean *f* produktu finantzarioan. (Adibidez, *Errenta*(70,3)=4 bidez adierazten da, 700 euro 3 produktuan inbertituz, %4 mozkina espero denez, 28 euroko onura lortuko dugula.) Oharra: garrantzitsua da jabetzea *f* produktu bakoitzeko mozkinaren portzentaia aldakorra dela hartan inbertitutako diru kopurupean, hala eta guztiz ere, errentagarriagoa da kopuru berdina behin soilik produktu jakin batean inbertitzeak, produktu berean zenbait inbertsio egitea baino. Hau da, balidn $d=d_1+d_2$, orduan $Errenta(d,f)>Errenta(d_1,f)+Errenta(d_2,f)$.

Programazio dinamikoaren teknika erabiliz diseina ezazu algoritmo bat zehazteko lor genezakeen onura maximoa D hamarreko euro bagenitu (suposatu $D\leq M$).

Aurreko algoritmoa egoki ezazu onura maximo hori lortzeko jakin dezagun, gainera, zeintzuk inbertsio egin behar diren.

31) Maite eta Patxi aktoreek hamaika opari jaso dituzte entzute handia izan duen telebistako serie batean egin duten interpretazio bikainagatik. *I* opari bakoitza biontzat igorritako kaxa baten barnean dago, haren P_i pisua soilik ezaguna izanik. Opari bakoitza zer den jakiteko, kaxa ireki behar da, eta horretarako astirik ez dutela eta, opari guztiak elkarren artean banatzea erabaki dute, biok pisu berdina batzen duten kaxa multzoa bat eskuratuz. Denbora luze baten ondoren, opari sorta erdibitzea ez dute lortu. Are gehiago, euren irizpidearekin opari kaxak banagarriak ote diren ere ez dakite. Tekla informatikariari laguntza eskatu diote, tankera honetako problemak ebazten trebea baita; baina Tekla guztiz lanpeturik dagoenez eta zu haren laguna zarenez, errebote gisa problema zureganaino iritsi da. Egizu algoritmo bat Maiteren eta Patxiren irizpidea erabiliz oparien banaketa egingarria den ala ez erabakiko.