Algoritmo jaleak
(Alg. Voraces - greedy algorithms)

R. Arruabarrena
LSI - UPV/EHU

Sarrera

- eguneroko bizitzan optimizazio problemak nonahi
- maiz enuntziatu sinpleak konputagailua bidezko ebazpenean denbora asko:
  - Backtracking
- praktikan aplikagarriagoak diren bi teknika:
  - hurbilketa bidezko soluzioak   Alg. Jaleak
  - soluzio optimoak   P. Dinamikoa

PD vs. Jalea

- P.D.:
  - soluzio optimoak kalkulatzen ditu
  - BAINA memoria memoria espazio estra + denbora ordena da
- JALEA:
  - sinplea, azkarra, espazio gutxi behar
  - BAINA ez du beti balio

Soluzio Jalea = Jale ZUZENA

- Soluzio jale zuzena: problemaren sarrera guztientzat soluziorik onena mugatzen duenean
  - Froga behar du izan
  - Ekintza hautaketa, Kruskal, Prim eta Dijsktra algoritmo jale zuzenak dira, euren frogak irakasgaiaren oinarrizko liburuan
- Sol. optimoaren hurbilketa jalea: Froga ez dugunean baina sarrera ia guztientzat soluzio onena kalkulatzen duenean.
  - kontrako adibideak topatzea zaila izan behar du
  - Honekin aski bada, konformatu behar. Bestela, P.D. Backtrack edo bibliografira jo

Alg. Jaleen prozesua

Soluzioa lortu arte errepika (edo hautagaiak agortu artean):

- une bakoitzean **aukera onena** egin
  - gelditzen diren hautagaietatik irabazi-etekin gehien ematen duena
- Aukeratutakoa hautagaia BEHIN tratatu:
  - onargarria bada soluzio partzialari gehitu
  - bestela baztertu

Soluzio jalearen helburua

- Optimizazio lokalak eginaz optimizazio globala lortzea
  - Lokala: gelditzen direnen artean aukera onena une bakoitzean. **Mokadu onena**
  - Globala: soluzioa zuzena izatea – optimoa
- Gure helburua:
  - Adibide zuzen batzuk ikasi eta euren adaptazioa beste problema batzuei
    - Bibliografian zuzen gehiago
  - Interesgarria denean soilik hurbilpen jaleak direnak egiten ikasi

Eskema jalea

```
function JALEA (H:Hautagai-multzoa)
                 return Hautagai-multzoa is
  SP: Hautagaien-multzoa;
begin

  MultzoHutsa(SP);
  while  not(SoluzioaDa?(SP)) and
             not(MultzoHutsaDa?(H))   do
      HautesleProzedura(H,x);         -- x hautatu jalea
      if  Osogarria?(Erantsi(SP,x))
      then  SP:= Erantsi (SP,x); end if;
  end loop;

  if  SoluzioaDa?(SP)   then return(SP)
  else MultzoHutsa(SP); return (SP); end if;
end JALEA;
```


Alg. Jaleen osagai komunak

1. hautagaien multzoa
   - Adib.: prozesa ditzakegun atazak, grafoko erpinak, arkuak edo ertzak, ...
2. SP: Soluzio Partziala
   - soluzio partziala jadanik aukeratu eta <u>onartu</u> diren hautagaien multzoa
   - Aurrekoaren azpimultzoa da
3. SoluzioaDa? hautagaien azpimultzo bat gure arazoaren soluzioa den ala ez erabakitzen duen funtzioa
   - nahiz eta soluzio hoberena ez izan

Alg. Jaleen osagai komunak

4. HautesleProzedura*:* oraindik aukeratu ez diren hautagaietatik soluzio onenaren bideratzailea den hautagaia aukeratuko duen prozedura

5. Helburu-funtzioa: soluzio bati dagokion balioa edo kostua itzultzen duena
   - Maximizatze / minimizatze funtzioa

6. Osogarria f.: aukeratutako azken hautagaia tarteko emaitzari erantsi garria den ala ez erabakitzen duena

Ekintza hautaketa ≡ baliabide konpartitua

Baliabide bera eskuratzearren lehiatzen duten eskaeren arteko azpimultzo bat identifikatu nahi da, non multzoaren kardinalitatea maximoa izanik, bertako eskaerak baliabidea esklusioan erabiltzen duten

- **Datuak**: baliabide bera eskuratzearren lehiatzen duten ekintza multzoa
  - N_EK = {1,...,N}.
  - *i* ekintza: *hi* hasiera-denbora; *bi* bukaera-denbora, *hi*≤*bi* betez
- **Problema**: baliabide baten erabilpen esklusiboa ekintzen artean banatu
- **Helburua**: zerbitzua lortzen duten eskaera kopurua maximatzea
- **Soluzioa**: denboran teilakatzen ez diren sarrerako datuen kardinalitate handieneko azpimultzo bat

Ekintza hautaketa. Prozedura hautesleak

a) Gutxien irauten duen ekintza aukeratu

b) Lehenen hasten dena

c) Denbora ardatzean gutxien teilakatzen dena

d) Lehenen amaitzen dena

f) Beranduen amaitzen dena

g) Beranduen hasten dena

…

Ekintza hautaketa. Prozedura hautesleak

a) Gutxien irauten duen ekintza aukeratu

Esperantza: Laburrena bada, gehiago sartuko dira eta horrela kardinalitate handiagoa.

Gezurra: B AbCa c

Jakinik :B ekintzaren hasiera denbora; b:ekintzaren amaiera denbora

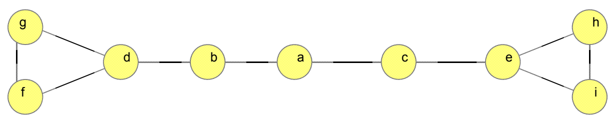
b) Lehenen hasten dena

Esperantza: Lehenago hasten bada, beste bat etorri aurretik dagoeneko martxan egongo da eta dagoeneko egiten/aprobetxatzen.

Gezurra: : A BbCc Dda

c) Denbora ardatzean gutxien teilakatzen dena

Esperantza: gutxirekin teilakatzen bada, hura soluzioan egoteak eragin txikia izango du (soilik teilakatzen diren haiengan), honela besteak (gehienak) sartzeko aukera gehiago dute



f) Beranduen amaitzen dena

Esperantza: Berandu bukatzen bada, seguruenik berandu hasiko dela, ez?
Gezurra: : A BbCc Dda

- P. hautesle onak soilik dira:
  d) Lehenen amaitzen dena
  g) Beranduen hasten dena
  Oharra: zuzentasun froga liburuan

- **Osogarria?** Baldin eta aukera berria orain arte aukeratutakoekin teilakatzen ez bada
  - J azkenekoz onartutako eskaera denean, eta I tratatzen ari garena:

    `BukaeraT(J) ≤ HasieraT(I)`

- **SoluzioaDa?** Ekoiztutako azpimultzoa beti izango da soluzioa:
  - teilakapenik ez du, eta
  - kardinalitate maximoa

Ekintza hautaketa. Algoritmoa (d prozedura hauteslea)

```
function EkintzaHauteslea (NEk : in EkintzenL)
                    return EkintzenL is
   NEkL: EkintzenL; I,J: Ekintza;
   SP: EkintzenL:= Hasieratu;
begin
   BukaeraTordenaGorakorraJarraituz(NEk, NEkL);
   LehenengoaL (NEkL, J); HondarraL(NEkL);
   ErantsiL (SP, J);
   while  not HutsaDaL?(NEkL)  loop
        LehenengoaL (NEkL, I); HondarraL (NEkL);
        if BukaeraT(J) ≤ HasieraT(I)
        then  J:= I;  ErantsiL (SP, J); end if;
   end loop;
   return SP;
end EkintzaHauteslea;
```

Analisia: $\Theta(n \lg n + n) = \Theta(n \lg n)$

Adibidea

1. “Lehenen amaitzen den eskaera” prozedura hauteslea integratua duen algoritmo jaleak zer itzuliko du ondorengo eskaerekin?

   {[8:30, 9:30), [12:00, 12:45), [11:30, 12:15), [11:00, 12:30), [10:00, 10:50) }

2. Eta “Beranduen hasten den eskara” prozedurakin?

Adibidea

1. "Lehenen amaitzen den eskaera" prozedura hauteslea integratua duen algoritmo jaleak zer itzuliko du ondorengo eskaerekin?

   { [8:30, 9:30), [12:00, 12:45), [11:30, 12:15), [11:00, 12:30), [10:00, 10:50) }

   { [8:30, 9:30), [10:00, 10:50), [11:30, 12:15), ~~[11:00, 12:30)~~, ~~[12:00, 12:45)~~**}**

2. Eta "Beranduen hasten den eskaera" prozedurakin?

   {[12:00, 12:45), ~~[11:30, 12:15), [11:00, 12:30),~~ **[10:00, 10:50), [8:30, 9:30) }**

Motxila 0/1 vs. zatiekin

- Datuak:
  - 1..n objektuak: p(i)= i objektuaren pisua eta b(i) i-ren balioa
  - Edu: Motxilaren edukiera
- Helburua 0/1: motxilaren edukiera gainditu gabe lor litekeen irabazi maximoa objektuak osorik sartuz
  - Beste bertsioa: **Objektu zatiak sar garri**

- Pareko beste enuntziatu batzuk:
  - 0/1: Tren bagoiak+ granito blokeak; furgonetak + paketeak
  - zatiekin: garraioak + kafe zakuak

Motxila. Prozedura hautesleak

a) Gehien balio duena:

Esperantza: Irabazi lokal maximoa ematen digu, daudenen artetik etekin handien ematen duena oraingoz, eta gero besteak gehituko dizkiogu.

Gezurra:	E=10	
p1=8	p2=4	p3=5
b1=15	b2=10	b3=10

b) Arinena:

Esperantza: Oso gutxi pisatzen duenez, gehiago sartuko zaizkigu eta ondorioz bakoitzaren etekina batuz irabazi gehiago lortuko dugu, ez?.

Gezurra:	E=10
p1=8	p2=4
b1=15	b2=10

Motxila. Prozedura hautesleak

c) Max{Balioa(I)/Pisua(I)}: edo parekoa Min{Pisua(I)/Balioa(I)}

Esperantza: Bien arteko erlazioa proportzioan irabazi onena ematen diguna aukeratzea dirudi.

Gezurra:		E=10
p1=7	p2=5	p3=5
b1=7	b2=4	b3=4

ONDORIOA:

Motxila 0/1 problemak EZ du ORAINGOZ soluzio jale zuzena. Frogarik ez da lortu

Aldiz, (c) p. hauteslea zuzena da motxila zatiekin kasurako.

Frogak existitzen du

Motxila **zatiekin**.

Osogarria: Sartzen den bitartean, osorik edo proportzio bat, baina edukiera gainditu gabe

SoluzioaDa?: Kasu honetan beti izango da soluzioa eraikitzen goazen azpimultzoa

Algoritmoa: aldagaiak
Pisua(I): I objektuaren pisua
Balioa(I): I objektuaren irabazia/prezioa/etekina,...
Propor(I):=Balioa(I)/Pisua(I);
Atzipena(I)=J; I.proportzio handiena J objektuak ematen du.


```
procedure MotxilaZatikin (E: in Integer; Pisua,Balioa: in
                         TaulaN; Irabazia: out Integer;
                                   Obj: out TaulaN)
   Proportzioa,AtzipenOrd: TaulaN;
   Irabazia,EspazioLibrea, In,Garrena:Integer;
begin
   HasieratuZeroraObj(Obj); ProporLortu(Pisua,Balioa, Propor);
   ProportzioarekikoBeheruntzOrdenatu (Propor,AtzipenOrd);
   Irabazia:= 0; EspazioLibrea:=E; Ind:=1;
   while  0<EspazioLibrea and  Ind<N+1  loop
     Garrena:=AtzipenOrd(Ind);
     if Pisua(Garrena) ≤ EspazioLibrea
     then  Obj(Garrena):=1;
           EspazioLibrea:= EspazioLibrea-Pisua(Garrena);
           Irabazia:=Irabazia+Balioa(Garrena);
     else  SartuBeharDa:= EspazioLibrea/Pisua(Garrena);
           Obj(Garrena):= SartuBeharDa; EspazioLibrea:= 0;
           Irabazia:=Irabazia+SartuBeharDa* Balioa(Garrena);
     end if;
     Ind:=Ind+1;
   end loop;
end MotxilaZatikin;
```


Adibidea

Motxila zatiekin algoritmo jalearen sarrerako datuak:
p=[7, 5, 8, 4, 8]   Edu= 15
b=[7, 5, 9, 6, 5]

ProportzioarenAraberaBeheruntzOrdenatu azpiprozesua: efektua adibide bidez:

Azpiprozesuaren sarrera datuak
Proportzioa:

AtzipenOrd:

EMAITZA   Obj =
Irabazia =
EspazioLibrea =

Adibidea

Motxila zatiekin algoritmo jalearen sarrerako datuak:
p=[7, 5, 8, 4, 8]   Edu= 15
b=[7, 5, 9, 6, 5]

ProportzioarenAraberaBeheruntzOrdenatu azpiprozesua: efektua adibide bidez:

Azpiprozesuaren sarrera datuak
Proportzioa: 1 1 1.13 1.5 0.63

AtzipenOrd: 4 3 2 1 5

EMAITZA   Obj = [0, 3/5, 1, 1, 0]
Irabazia = 6+ 9+ (3/5)
EspazioLibrea = 0

H.Z.M. - Kruskal

Grafoko erpin guztiak konektatzen dituen eta haien pisuen batura minimizatzen duen den ertzen multzoa mugatzea duen algoritmoa idatzi nahi da; hots, grafo (konexu, ez-zuzendua eta pisudunaren) **H**edapen **Z**uhaitz **M**inimoa kalkulatzea

Datuak:

- Pisuak positiboak izan behar dute
- **Kruskal**en soluzioak: grafoa ertz pisudunen **zerrenda** bidez adierazia behar du

- **Hautesle prozedura:** pisu txikieneko ertza aukeratu.
- **Osogarria**: Ziklorik ez du gehitzen *ERT Soluzio Partzialean*
- **SoluzioaDa?**: Zuhaitza bada SP

Froga liburuan kontsultagarri

Kruskalen algoritmoaren hurbilketa

```
algoritmoa KRUSKAL (G=<Erpinak, Ertzak,Pisuak) ERT: Ertz-
  multzoa
  ...
hasi
  L:= PisuenGoranzkoOrdenaJarraituzSailkatu (Ertzak);
  P := ErpinKopurua (Erpinak);
  MultzoHutsaErt (ERT);
  errepika  (Ertz_kopurua(ERT) ≠ P-1) hasi
      Pisu_txikieneko_ertza_aukeratu(L,(x,y));
      Aukeratutako_ertza_kendu(L,(x,y));
      baldin not Ziklorik_eransten_du?(ERT U {(x,y)})
      orduan ERT:= ERT U {(x,y)};
      bukatu baldin;
  bukatu errepika;
bukatu KRUSKAL
```


H.Z.M. – Kruskalen algoritmoa

```
procedure KRUSKAL(G: in GRAFOA; SErt: out Ertz_multzoa)  is
   OsKonexuak: PPartiketaMota;
   P,SErtzKop: Integer;
begin
   L:= PisuenGoranzkoOrdenaJarraituzOrdenatu(ERTZAK(G));
   P := ErpinKop (G); SErtzKop:=0; MultzoHutsaErt (SErt);
   pMultzoHaseratuBakoitzaErpinEzberdinBatekin(OsKonexuak);

   while  (SErtzKop ≠ p-1)  loop
        KontsideratuEzDenPisuTxikienekoErtza (L,  (x,y));

        XBarne:=BILATU3(OsKonexuak, X);
        YBarne:=BILATU3(OsKonexuak, Y);

        if Xbarne≠Ybarne then
           BATERATU3(OsKonexuak, XBarne, YBarne);
           ErantsiErt(SErt, (x,y));
           SErtzKop:= SErtzKop+1;
        end if;
   end loop;
end KRUSKAL;
```


- Kruskal. Analisia
  - Hasieraketak: $\Theta(a \lg a + p + p) = \uparrow = \Theta(a \lg p)$
    $p-1 \le a \le (p(p-1))/2 \rightarrow \Theta(\lg a) = \Theta(\lg p)$
  - Begizta: kasu txarrenean ordenazio zerrendako azkeneko ertza gehitzen da Sert multzora
    - [baterat3,bilatu3]^n $\in \Theta$(n lgn)
    - $\underbrace{\text{2a aldiz Bilatu3 + (p-1) aldiz Bateratu3}}$

      $\in \Theta((2a+p) \lg (2a+p))$
      $= \Theta(a \lg a) = \Theta(a \lg p)$
  - Beraz, baturaren erregela bi agindu blokeei aplikatuz: **$\Theta(a \lg p)$**

H.Z.M. – Prim

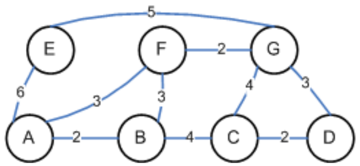
Grafoaren **H**edapen **Z**uhaitz **M**inimoaren kalkulua

Datuak + helburua:

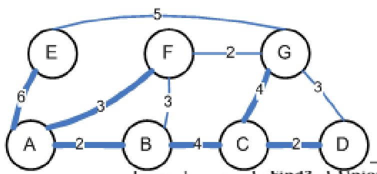
- Kruskalen problema bera prozedura hautesle desberdinez ebatzia
- Grafoa **auzokidetza matrize** bidez adierazia

Algoritmoa: aldagaiak

- ERP: erpinen multzo fiktizioa. SErt multzoko ertzetan dauden grafoko erpinez osatua
- i ∊ Erpinak-ERP
- Auzokide(i): *i* erpinetik ERP-ekoa den eta hurbilen duen erpina
  - Hau da, (i,AUZOKIDE(i)) ertzak momentu horretako SErt multzoa eta *i* erpina modu merkeenean konektatzen dituen ertza



Bira	ertza	Find3 etiketak	Union3 Bai-Ez	H.Z.M.ri gehitu	Partiketa a	b	c	d	e	f	g
0											
1	(__,__)	__,__									
2	(__,__)	__,__									
3	(__,__)	__,__									
4	(__,__)	__,__									
5	(__,__)	__,__									
6	(__,__)	__,__									
7	(__,__)	__,__									
8	(__,__)	__,__									
9	(__,__)	__,__									

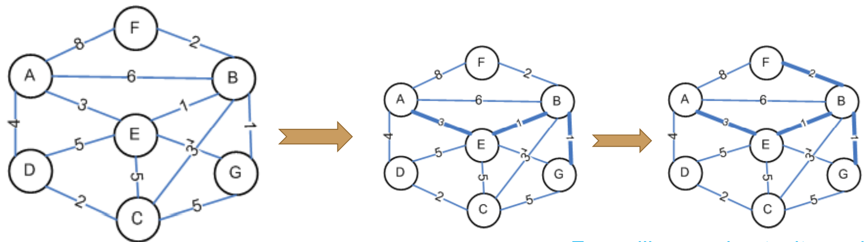


Bira	ertza	Find3 etiketak	Union3 Bai-Ez	H.Z.M.ri gehitu	Partiketa a	b	c	d	e	f	g
0					0	0	0	0	0	0	0
1	(C, G)	C, G	BAI	(C, G)	0	0	-1	0	0	0	C
2	(A, B)	A, B	BAI	(A, B)	-1	A	-1	0	0	0	C
3	(C, D)	C , D	BAI	(C, D)	-1	A	-1	C	0	0	C
4	(A, F)	A, F	BAI	(A, F)	-1	A	-1	C	0	A	C
5	(D, G)	C, C	EZ		-1	A	-1	C	0	A	C
6	(B, C)	A, C	BAI	(B, C)	-2	A	A	C	0	A	C
7	(B, G)	A, A	EZ		-2	A	A	C	0	A	C
8	(B, F)	A, A	EZ		-2	A	A	C	0	A	C
9	(A, E)	A, E	BAI	(A, E)	-2	A	A	C	A	A	C

H.Z.M. – Primen algoritmoa

h.z.m.-ren eraikuntza hedatuz egiten da, bertako erpinak ERP multzo fiktizioa osatuz.

Hautesle prozedura: (Erpinak-ERP) eta ERP multzoko erpinak konektatzen dituzten ertzen artetik pisu txikieneko ertza aukeratu



Froga liburuan kontsultagarri

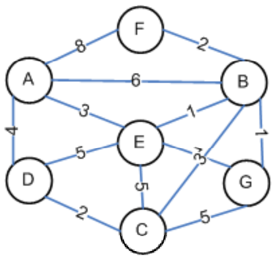
H.Z.M. – Primen algoritmoa

Algoritmoa: aldagaiak

- Auzokide(i):
  - i ∈ Erpinak-ERP; hots h.z.-ean oraindik ez dagoen erpina
  - espresioak *i* erpinetik ERP-ekoa den eta hurbilen duen erpina metatzen du; hots, (i, Auzokide(i(i)) ertzak momentu horretako SErt multzoa eta *i* erpina modu merkeenean konektatzen dituena da.
- PisuMin(i):
  - (i, Auzokide(i)) ertzaren pisua
  - *i* ∈ERP → PisuMin(i) = -1


```
procedure  PRIM (G: in Matrizea; SErt: out ErtzMultzoa) is
begin
  MultzoHutsaErt (SErt);
  for  K   in  G´First(1) +1..G'Last(1)   loop
       Auzokide(K)  := 1; PisuMin := G(K,1);
  end loop;

  for Ind in  G´First(1)..G'Last(1)-1   loop
    Min := System.MAX_INT;
    for J in G´First(1)+1..G'Last(1)  loop
      if 0≤PisuMin(J)<Min then Min:= PisuMin(J); K:= J;end if;
     end loop;

     ErantsiErt(SErt,(K,Auzokide(K)); PisuMin(K):= -1;
     for J in G´First(1)+1..G'Last(1)  loop
       if G(K,J)<PISU_MIN(J)
       then PisuMin(J):= G(K,J); Auzokide(J):= K; end if;
     end loop;
   end loop;
end PRIM;
```

$\Theta(p^2)$

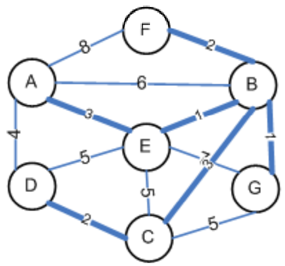


Iter.	Erpina	Ertz	Pisua		A	B	C	D	E	F	G
0	A			PisuMin							
				Auzokide							
1		(,)		PisuMin							
				Auzokide							
2		(,)		PisuMin							
				Auzokide							
3		(,)		PisuMin							
				Auzokide							
4		(,)		PisuMin							
				Auzokide							
5		(,)		PisuMin							
				Auzokide							
6		(,)		PisuMin							
				Auzokide							

Kostua Orora

Iter.	Erpina	Ertz	Pisua		A	B	C	D	E	F	G
0	A			PisuMin		6	∞	4	$\boxed{3}$	8	∞
				Auzokide		A	A	A	$\boxed{A}$	A	A
1	E	(A,E)	3	PisuMin		$\boxed{1}$	5	4	-1	8	7
				Auzokide		$\boxed{E}$	E	A	A	A	E
2	B	(E,B)	1	PisuMin		-1	3	4	-1	2	$\boxed{1}$
				Auzokide		E	B	A	A	B	$\boxed{B}$
3	G	(B,G)	1	PisuMin		-1	3	4	-1	$\boxed{2}$	-1
				Auzokide		E	B	A	A	$\boxed{B}$	B
4	F	(B,F)	2	PisuMin		-1	$\boxed{3}$	4	-1	-1	-1
				Auzokide		E	$\boxed{B}$	A	A	B	B
5	C	(B,C)	3	PisuMin		-1	-1	$\boxed{2}$	-1	-1	-1
				Auzokide		E	B	$\boxed{C}$	A	B	B
6	D	(C,D)	2	PisuMin							
				Auzokide							

Kostua Orora $\boxed{12}$



Kruskal vs Prim

- Kruskal $\in\Theta$(a lg a)   eta Prim $\in\Theta(p^2)$

- Kasuak:
  - Grafoak ertz asko → a <≈ (p (p-1)/2)

    Kruskal $\in\Theta(p^2$ lg p) izango da (Θ(lg p)=Q(lg a))

    Prim-en hobea

  - Grafoak ertz gutxi → a ≈ n

    Kruskal $\in\Theta$(p lg p), hortaz Prim baino eraginkorra

Distantzia minimoak - Dijkstra

G grafo zuzendua eta pisuduna: G = <Erpinak, Arkuak, Pisuak>.
Erpin bat abiapuntutzat hartuko dugu.
Helburua: abiapuntu erpinetik grafoko beste erpinetara doazen bide motzenak mugatzea

- H hautagai erpinen multzoa.
- E erpin hautatu eta onartuen multzo fiktizioa.
- **Bide berezia:** abiapuntu-erpinetik beste erpin batera doan bidea eta E-ko erpinez osatuta dagoena soilik.
- Bide berezi motzenak bakarrik kontuan hartzeak bide motzenak ekoizten dituztela frogagarria da (ikus liburua)

- Kalkuluak erraztearren :
  - grafoko erpinak: {1,2, .. , P}
  - abiapuntu-erpina: 1
  - Auzokidetza matrizea

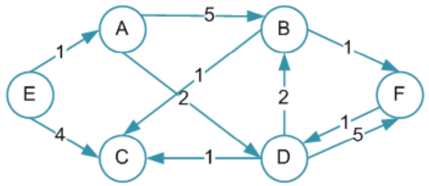
    G(i,j) ≥ 0 -> $\exists$(i,j)$\in$G eta bere pisua da

    G(i,j) = ∞ -> $\neg\exists$(i,j)$\in$G

Prozesua: Froga liburuan kontsultagarri

- Begizta jalearen bira bakoitzean:
  - bide berezi motzena aukeratzen da, bide berezi motzena bide motzena bihurtuz.
  - E-koa ez den erpin berria E-ri gehitzen zaio (H-tik kenduz),
  - erpin berria gehitzeagatik sortu diren bide berezi berriak aurrekoak baino motzagoak badira erregistratzen dira.

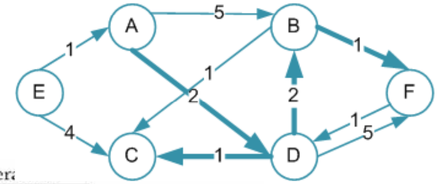

```
procedure  DIJKSTRA ( G: in GRAFOA;  D: out TAULA) is
   P: Integer:=G'LENGTH(1);
begin
   HASIERATU2_P(G,H);
   for K in  G'FIRST(1)+1..G.'LAST(1) loop D(K):= G(1,K);
   end loop;

   for  I in  P-2  loop
      GertuenDagoenErpina(H,D,X);          ∈Θ(p²)
      HautagaietatikKendu(H,X);            ∈Θ(p²)
      HK:= H;
      for J in 1..ErpinKopurua(H) loop
       Y:= LehenengoaL(HK);
       AurreraEgin(HK);
       if D(Y)>D(X)+G(X,Y) then D(Y):= D(X)+G(X,Y); end if;
      end loop;                  (p-2) + (p-3)+ ... + 1 ∈Θ(p²)
   end loop;
end DIJKSTRA ;                             Θ(p²)
```



Distantzia taularen hasieratzea:

A	B	C	D	E	F

	Harrapatutako erpina	Distantzia taularen eguneraketa: A B C D E F
1 bira		
2 bira		
3 bira		
4 bira		



Distantzia taularen hasier:

A	B	C	D	E	F
X	5	∞	2	∞	∞

	Harrapatutako erpina	A	B	C	D	E	F
		Distantzia taularen eguneraketa:					
1 bira	D	X	4	3	2	∞	7
2 bira	C	X	4	3	2	∞	7
3 bira	B	X	4	3	2	∞	5
4 bira	F	X	4	3	2	∞	5