
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 onartu 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. **Hautesle-Prozedura**: 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

H.Z.M. –Minimum Spanning Tree

Grafoko erpin guztiak konektatzen dituen ertzen multzoa, haien pisuen batura minimoa izanik, mugatuko duen algoritmoa idatzi nahi da; hots, grafo konexu, ez-zuzendu eta pisudunaren **H**edapen **Z**uhaitz **M**inimoa kalkulatzeko

Aplikazioak:

- Konputagailu sareak, konexio sareak, telekomunikazio sareak, garraio sareak, sare elektrikoak, ur banaketa sareak,...

Bi soluzio ospetsu:

- Prim, Kruskal

H.Z.M. - Kruskal

Datuak:

- ❑ Grafoa ez-zuzendu konexua izan behar du
 - ❑ Pisuak positiboak
 - ❑ **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](#)

R. Arruabarrena

11

Kruskalen algoritmoaren hurbilketa

```
algoritmoa KRUSKAL (G=<Erpinak,Ertzak,Pisuak>) ERT: Ertz-  
multzoa  
...  
hasi  
  L:= PisuGoranzkoOrdenaJarraituzSailkatu (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
```

R. Arruabarrena

12

H.Z.M. – Kruskalen algoritmoa

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

R. Arruabarrena

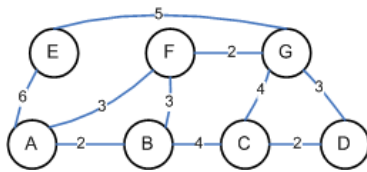
13

■ Kruskal. Analisia

- Hasieraketak: $\Theta(a \lg a + p + p) = \uparrow = \Theta(a \lg p)$
 $p-1 \leq a \leq (p(p-1))/2 \rightarrow \Theta(\lg a) = \Theta(\lg p)$
- Begizta: kasu txarrean ordenazio zerrendako azkeneko ertza gehitzen da SErt multzora
 - $[baterat3, bilatu3]^n \in \Theta(n \lg n)$
 - $$\underbrace{2a \text{ aldiz Bilatu3} + (p-1) \text{ aldiz Bateratu3}}_{\substack{\in \Theta((2a+p) \lg (2a+p)) \\ = \Theta(a \lg a) = \Theta(a \lg p)}}$$
- Beraz, baturaren erregela bi agindu blokeei aplikatuz: $\Theta(a \lg p)$

R. Arruabarrena

14



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

15

H.Z.M. – Kruskalen algoritmoa*

procedure KRUSKAL(G: **in** GRAFOA; SErt: **out** Ertz_multzoa) **is**

...

Begin

```

(P,A) := ErpinKopErtzKop (G);
SErtzKop:=0; MultzoHutsaErt (SErt);
MinMetaEraiki (MinMeta[1..A], ERTZAK_ARR(G));
PartiketaHasieratu (OsKonexuak);

while (SErtzKop ≠ P-1) loop
  (x,y) := ErroaLortuEtaMinMetaBerritu (MinMeta);
  XBarne:=BILATU4 (OsKonexuak, X);
  YBarne:=BILATU4 (OsKonexuak, Y);

  if Xbarne≠Ybarne then
    BATERATU4 (OsKonexuak, XBarne, YBarne);
    ErantsiErt (SErt, (x,y));
    SErtzKop:= SErtzKop+1;
  end if;
end loop;
end KRUSKAL;

```

O(???)