

Bektoreko elementuen ordenazioa

- Burbuilaren metodoa
- Hautaketa bidezko metodoa
- Txertaketa bidezko metodoa.

Metodo horien portaera bisualki aztertzeko, ikusi

<http://math.hws.edu/eck/js/sorting/xSortLab.html>

<https://www.youtube.com/watch?v=X8SGNMPzjDQ>

Burbuila deritzon algoritmoa

Oso-bektore bat emanda, bektoreko osagaiak *Burbuila* deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa.

Burbuila metodoaren deskribapena:

1. Ondoz ondoko bi elementu sekuentzialki konparatuz eta komeni denean trukatzuz, elementurik handiena bere tokian jartzen da.
2. Prozesu hori errepikatu egiten da, harik eta elementu guztiak ordenatu arte.
3. Konparazio eta trukaketa asko egin behar izaten dira.

```
procedure Ordenatu_Burbuilaz (B: in out Oso_Bektore) is
-- Aurre:
-- Post: Bko elementuak ordenatuta daude
--       txikienetik handienara

begin
  for I in reverse B'First+1..B'Last loop
    -- B(I+1..B'Last) bektoreko osagaiak ordenatuta daude
    -- eta B bektoreko handienak dira
    Korritu_Elementuak_Handiena_Jasoz (B, B'First, I);
  end loop ;
end Ordenatu_Burbuilaz ;

procedure Korritu_Elementuak_Handiena_Jasoz
  (B: in out Oso_Bektore;
   I_Hasiera, I_Bukaera: in Integer) is
-- Aurre:
-- Post: B(I_Hasiera..I_Bukaera) bektoreko elementu handiena
--       I_Bukaera posizioan dago.

begin
  for K in I_Hasiera..I_Bukaera - 1 loop
    if B(K) > B(K+1) then
      Balioak_Trukatu (B(K), B(K+1)) ;
    end if ;
  end loop;
end Korritu_Elementuak_Handiena_Jasoz ;
```

Hautaketa bidezko algoritmoa

Metodoaren deskribapena:

1. Aurkitu bektoreko balio txikiena.
2. Trukatu balio txikien hori lehenengo posizioko balioarekin.
3. Errepikatu bi aurreko pausoak zerrendako gainontzeko elementuekin (bigarrenetik aurrera)
4. Aldaera: balio handienekin ere egin daiteke, balio handienak bektorearen bukaeran jarritz.

```
procedure Ordenatu_Hautaketaz (B: in out Oso_Bektore) is
-- Aurre:
-- Post:  B bektoreko elementuak ordenatuta daude
--        txikienetik handienara
begin
    for I in B'First..B'Last loop
        -- B(B'First..I) bektoreko osagaiak ordenatuta daude
        -- eta B bektoreko txikienak dira
        Bilatu_Minimoaren_Posizioa (B, I, B'Last, Pos);
        Balioak_Trukatu (B(I), B(Pos)) ;
    end loop ;
end Ordenatu_Hautaketaz ;

procedure Bilatu_Minimoaren_Posizioa
    (B: in Oso_Bektore;
     I1, I2: in Integer;
     Minimoaren_Posizioa: out Integer) is
-- Aurre: I1 eta I2 B bektorearen indize posibleak dira.
-- Post:  B(I1..I2) sekzioko txikiena B(Minimoaren_Posizioa) da.
begin
    Minimoa :=Integer'Last ;
    for K in I1..I2 loop
        if B(K) < Minimoa then
            Minimoa := B(K);
            Minimoaren_Posizioa := K;
        end if ;
    end loop;
end Bilatu_Minimoaren_Posizioa ;
```

Txertaketa bidezko algoritmoa

Metodoaren deskribapena:

1. Ordenatu gabeko bektore-zatiko hurrengo elementua hartzen da.
2. Elementu hori ordenatutako bektore-zatian txertatzen da, dagokion posizioan. Eginkizun honek elementuak desplazatzea eskatzen du.
3. Errepikatu bi aurreko pausoak ordenatu gabeko elementurik geratzen ez den arte.

```
procedure Ordenatu_Txertaketaz ( B : in out Oso_Bektore ) is
  -- Post: B bektoreko lelementuak ordenatuta daude
  -- txikienetik handienera
  Unekoa, Posizioa: Integer;
begin
  for I in B'First+1..B'Last loop
    Unekoa:=B(I);
    Desplazatu(B,Unekoa,I,Posizioa);
    B(Posizioa):=Unekoa;
  end loop;
end Ordenatu_Txertaketaz;

procedure Desplazatu
  (B: in out Oso_Bektore;
   E, P: in Integer;
   Posizioa: out Integer) is
  -- Aurre: B(B'First..P-1) bektoreko elementuak ordenatuta daude
  -- gorantz.
  -- E=B(P).
  -- Post: B(B'First..P) bektoreko elementuak ordenatuta daude gorantz.
  -- B(P+1..B'Last) bektorea ez da aldatu.
  J: Integer;
  Topatua: Boolean;
begin
  J:=P-1;
  Topatua:=False;
  while J>=1 and Topatua = False loop
    if B(J)>E then
      B(J+1):=B(J);
      J:=J-1;
    else
      Topatua:=True;
    end if;
  end loop;
  Posizioa:=J+1;
end Desplazatu;
```