

DEA-II EKAINEKO DEIALDIA (2010-v-31) - E taldea

Deiturak	1	2	3	4	5	

1.- (2 pt) Ondorengo algoritmoa azter ezazu, jakinik $Sortua(x,y)$ prozedura $O(1)$ ordenakoa dela:

```
proc Azterketa(i, j)
  if (i=j) then Sortua(i, j)
  else e:=(i+j)/2
       k:=(j-i)/4
       Azterketa (i, E)
       Azterketa (i+k, E+k)
       Azterketa (E+1, j)
       Konbinatu(i, j)
  end if
end Azterketa

proc Konbinatu (i, j)
  for a in i .. j loop
    for b in i ..j loop
      Sortua(a, b)
    end loop
  end loop
end loop
```

2.- (2 pt) Izan bedi $T(1..k)$ orora n osagai k zerrendetan jasotzen dituen egitura. $T(i)$ ($1 \leq i \leq k$) zerrenda bakoitza goranzko ordena jarraituz ordenaturik dago eta zerrenda egitura bezala manea liteke ondorengo metodoak erabiliz:

```
// Z-ko lehenengo elementua X-en itzultzen du, zerrendatik kenduz
removeFirst(Z: in out ZerrendaI; X: out Integer);

// Z-ko azkeneko elementua X-en itzultzen du, zerrendatik kenduz
removeLast(Z: in out Zerrenda; X: out Integer);

// Z zerrendak osagairik ez badu true itzultzen du; eta false bestela
isEmpty(Z: in out Zerrenda) return boolean;
```

$T(1..k)$ egiturako zerrendetako osagai guztiak $A(1..n)$ taulara eramaten dituen algoritmoa idatz ezazu $\forall i:1..n-1.A(i) \leq A(i+1)$ efektua lortuz. Idatzitako algoritmoa $O(n \lg k)$ ordenakoa izan behar du.

3.- (2 pt) Hiri bateko kaleak adierazteko grafo bat daukagu, non bidegorriak diren kaleak markatuak dituen.

Hiriko bi bidegurutze emanik, batetik bestera **soilik bidegorriak** erabiliz joan gitezkeen ala ez kalkulatu duen algoritmoa idatz ezazu. Algoritmoak ezingo du ez grafoa eraldatu ez eta kopiatu ere (nahiz, kopia partziala izan). Algoritmoaren analisia egizu.

4.- (2 pt) $T(1..n,1..n)$ osokoen taula dugu (1,1) posiziotik (n,n) posizioraino doazen bideen artetik, bideko posizioetako balioen batura **minimoa** duen **bidearen kostua** kalkulatzeko programazio dinamikoaren teknika erabil ezazu. Taulako posizio jakin batetik soilik haren eskuinean ala behean dagoen aldameneko posiziora mugitu gaitzke. Algoritmoa azter ezazu.

5.- (2 pt) K euro sakelan izanik azokara goaz. Honekin batera eros ditzakegun m produktuen zerrenda daramagu. i produktu bakoitzeko ($1 \leq i \leq n$) p_i prezioa eta e_i erabilgarritasun balioa (biak osoko positiboak) ezagunak ditugu.

Produktu bakoitzeko gehienez 3 produktu eros ditzakegu. Gainera, eguneko eskaintzari esker, produktu beraren bigarren unitatearen salneurria 2 euro gutxiago kostatuko zaigu eta hirugarren unitatea, aldiz, 3 euro gutxiago.

Erosketaren erabilgarritasuna erositako produktu bakoitzaren erabilgarritasunaren batura dela jakinik, Backtrack algoritmo bat idatz ezazu erabilgarritasuna maximoa duen erosketa mugatzeko eta produktu bakoitzeko erosi behar diren ale kopurua zehaztuko duena.

Algoritmoa egin aurretik irudika ezazu esplorazio zuhaitza identifikatzeko zeintzuk diren adarketak, entsaioen hedapenak, entsaio onargarriak, hostoak, soluzioak...

Positiboki baloratuko dira: Esplorazio zuhaitzaren diseinu on bat egitea eta kodeketak beharrezkoak ez diren adarrak kimatzea.

DEA-II EKAINENKO DEIALDIA (2010-v-31) - E taldea
SOLUZIO BAT

1.- (2 pt) Ondorengo algoritmoa azter ezazu, jakinik $Sortua(x,y)$ prozedura $O(1)$ ordenakoa dela:

```
proc Azterketa(i, j)
  if (i=j) then Sortua(i, j)
  else e:=(i+j)/2
       k:=(j-i)/4
       Azterketa(i, E)
       Azterketa(i+k, E+k)
       Azterketa(E+1, j)
       Konbinatu(i, j)
  end if
end Azterketa

proc Konbinatu(i, j)
  for a in i .. j loop
    for b in i .. j loop
      Sortua(a, b)
    end loop
  end loop
end loop
```

Sol:

$$T(1)=\Theta(1)$$

$$\begin{aligned} T(n) &= 3 T(n/2) + n^2 \\ &= 3 (3 T(n/2^2) + (n/2)^2) + n^2 = 3^2 T(n/2^2) + 3 n^2/4 + n^2 \\ &= 3 (3^2 T(n/2^3) + (n/2^2)^2) + 3 n^2/4 + n^2 = 3^3 T(n/2^3) + 3^2 n^2/4^2 + 3 n^2/4 + n^2 \\ &= 3^i T(n/2^i) + n^2 \sum_{j=0}^{i-1} 3/4 = 4 n^2 - 3 n^{lg 3} \in \Theta(n^2) \end{aligned}$$

2.- (2 pt) Izan bedi $T(1..k)$ orora n osagai k zerrendetan jasotzen dituen egitura. $T(i)$ ($1 \leq i \leq k$) zerrenda bakoitza goranzko ordena jarraituz ordenaturik dago eta zerrenda egitura bezala maneia liteke ondorengo metodoak erabiliz:

```
// Z-ko lehenengo elementua X-en itzultzen du, zerrendatik kenduz
removeFirst(Z: in out ZerrendaI; X: out Integer);

// Z-ko azkeneko elementua X-en itzultzen du, zerrendatik kenduz
removeLast(Z: in out Zerrenda; X: out Integer);

// Z zerrendak osagairik ez badu true itzultzen du; eta false bestela
isEmpty(Z: in out Zerrenda) return boolean;
```

$T(1..k)$ egiturako zerrendetako osagai guztiak $A(1..n)$ taulara eramaten dituen algoritmoa idatz ezazu $\forall i:1..n-1. A(i) \leq A(i+1)$ efektua lortuz. Idatzitako algoritmoa $O(n \lg k)$ ordenakoa izan behar du.

Sol:

```
proc MergeK (T: in array (1..k) of ZerrendaI; A: out array (1..n) of
Integer ) is
...
Begin
  for i in 1..k loop
    RemoveFirs(T(i), x1);
    M(i):=(x1,i);          -- balioa eta bere zerrenda bikotea metaratu
  end loop;
  MinMetaEraiki (M(1..k));
  for i in 1..n-1 loop
    ErroaLortu(M, (x1,z));
    A(i):=x1;
    if (not isEmpty(z))
    then RemoveFirs(T(i), x1);
        M(1):=(x1,z);
        Hondoratu(M,1);  -- Hondoratu erroa
    else AzkenaErrora(M);
        Hondoratu(M,1);
    end if;
  end loop;
  A(n):=M(1);
end;
```

3.- (2 pt) Hiri bateko kaleak adierazteko grafo bat daukagu, non bidegorriak diren kaleak markatuak dituen.

Hiriko bi bidegurutze emanik, batetik bestera **soilik bidegorriak** erabiliz joan gitezkeen ala ez kalkulatu duen algoritmoa idatz ezazu. Algoritmoak ezingo du ez grafoa eraldatu ez eta kopiatu ere (nahiz, kopia partziala izan). Algoritmoaren analisia egizu.

Sol:

```
procedure GrafoaKorritu (G1: in GrafoaAZ; Hasi,Bukatu: in Erpina;
                        Dago: out Boolean) is
  Aztertua: constant Boolean := True;
  Bisitak: Taula(ErpinKopEsparrua):= (Others=> not (Aztertua));
  BGDago: Boolean;
begin
  BGDago:=False; SK(G1, Hasi); Dago:= BGDago;
end GrafoaKorritu;

procedure SK (G: in Grafoa; A: in Erpina) is
  AAuzokideenKopia: ErpinenL; Lehena: Erpina;
begin
  AAuzokideen_Kopia:= Auzokideak(G, A);
  while not (HutsaDaL? (AAuzokideenKopia)) loop
    Lehena:= LehenengoaL(AAuzokideenKopia);
    Hondarra(A_AuzokideenKopia);
    if (lehena= buka) then BGDago:= true;
    else if (not Markatua_Dago(Bisitak, Lehena) and
            GorriaDaErtza(G1, (A,Lehena)) )
    then
      Marka_Ipini(Bisitak, Lehena, Aztertua);
      S_K(G, Lehena);
    end if;
  end loop;
end S_K;
```

4.- (2 pt) $T(1..n,1..n)$ osokoen taula dugu (1,1) posiziotik (n,n) posizioraino doazen bideen artetik, bideko posizioetako balioen batura **minimoa** duen **bidearen kostua** kalkulatzeko programazio dinamikoaren teknika erabil ezazu. Taulako posizio jakin batetik soilik haren eskuinean ala behean dagoen aldameneko posiziora mugi gaitzke. Algoritmoa azter ezazu.

Sol: $BM(i, j)$: (1,1) posiziotik (i,j) posizioraino bide minimoaren kostua

$$BM(i, j) = \min \{ T(i, j) + BM(i-1, j), T(i, j) + BM(i, j-1) \} \quad i, j > 1$$

$$= T(i, j) + BM(i-1, j) \quad j=1, i > 1$$

$$= T(i, j) + BM(i, j-1) \quad i=1, j > 1$$

$$BM(1, 1) = T(1, 1)$$

5.- (2 pt) K euro sakelari izanik azokara goaz. Honekin batera eros ditzakegun m produktuen zerrenda daramagu. i produktu bakoitzeko ($1 \leq i \leq n$) p_i prezioa eta e_i erabilgarritasun balioa (biak osoko positiboak) ezagunak ditugu.

Produktu bakoitzeko gehienez 3 produktu eros ditzakegu. Gainera, eguneko eskaintzari esker, produktu beraren bigarren unitatearen salneurria 2 euro gutxiago kostatuko zaigu eta hirugarren unitatea, aldiz, 3 euro gutxiago.

Erosketaren erabilgarritasuna erositako produktu bakoitzaren erabilgarritasunaren batura dela jakinik, Backtrack algoritmo bat idatz ezazu erabilgarritasuna maximoa duen erosketa mugatzeko eta produktu bakoitzeko erosi behar diren ale kopurua zehaztuko duena.

Algoritmoa egin aurretik irudika ezazu esplorazio zuhaitza identifikatzeko zeintzuk diren adarketak, entsaioen hedapenak, entsaio onargarriak, hostoak, soluzioak...

Positiboki baloratuko dira: Esplorazio zuhaitzaren diseinu on bat egitea eta kodeketak beharrezkoak ez diren adarrak kimatzea.

Sol:

Deskontua(0..3): eguneko eskaintza aplikatuz aplikatuko den deskontua erositako unitate totalari. 0 eta 1 ale erosiez gero 0, 2 unitate erosiez gero, solik bigarrenari, hortaz 2, eta 3 unitate erosiez gero, 2-ko murrizketa bigarrenari eta 3koa hirugarrenari, beraz 5 orora.

Erabiliko diren aldagaien zehaztapena:

$SP[X_1, \dots, X_i]$: j produktu bakoitzeko, non $1 \leq j \leq i$ den, zenbat ale erosi behar diren

$$SPB: \quad SP \text{ soluzio partzialaren kostua: } SPB = \sum (SP(j) * P(j) - \text{Deskontua}(SP(j)))$$

SPE: SP soluzio partzialaren erabilgarritasuna: $SPE = \sum(SP(j)*E(j))$

SOPTE eta SOPT: Erabilgarritasun optimoaren balioa eta zein erosketa antolaketa (SP) zehatzak ematen due erregistratzeko

```
procedure azoka (K: in integer; p,e: in array (1..N) of Integer;
  Erosketa: out array (1..N) of Integer;
  Erabilgarritasuna: out Integer) is
  Deskontua: array (1..N) of Integer:= (others=>0);
  SOPT, SP: array (1..N) of Integer;
  SOPTE, SPE, SPB: integer;
begin
  Deskontua(2):=2; Deskontua(3):=5;
  SOPTE:= 0;
  Azoka_Back01(1,0,0);
  Erosketa:= SOPT;
  Erabilgarritasuna:= SOPTE;
end;

procedure Azoka_Back01 (i,SPB,SPE)
Begin
  If (K=SPB) then      if SPE > SOPTE then    SOPTE := SPE;
                        SP(i..n):=ZerozOsatu; SOPT:= SP;
                        end if;
  elsif (i=n+1) then if SPE > SOPTE then    SOPTE := SPE; SOPT:= SP
                        end if;

  else for A in 0.. 3 loop
    if (SPB+ A*pi - Deskontua(A) ≤ K)
    then SP(i):= A;
        Azoka_Back01 (i+1, SPB+A*b(i)- Deskontua(A), SPE+A*ei);
    end if;
  end loop;
  end if;
end;
```