

DEA-II IRAILEKO DEIALDIA (2010-ix-1) - E taldea

Deiturak	1	2	3	4	5	

1.- (2 pt) Ondoren datorkizun $\text{BatuLaginak}(V, a, b)$ algoritmoak $V[a..b]$ bektoreko zenbait osagaien batura kalkulatu du. Algoritmoaren exekuzio ordena azter ezazu.

```
function BatuLaginak(V, a, b) return Zenbakia;
if b-a+1 ≤ 3 then return V[(a+b) div 2];
else y := (3*a+b) div 4;
      z := ((a+3b) div 4)+1;
return V[(a+b) div 2] + BatuLaginak(V, a, y) + BatuLaginak(V, z, b);
end if;
end BatuLaginak;
```

Aholkua: algoritmoaren funtzionamendua aztertzeko adibideren bat erabiltzen baduzu, 30 edo tamaina handiagoko sarrerak erabil itzazu.

Lortu duzun denbora-kostuaren funtzioa ondorengo multzo bakoitzeko elementua den ala ez adieraz ezazu:

$$\begin{array}{lll} O(\lg n) : & o(n) : & \Theta(\sqrt{n}) : \\ \Theta(\lg n \cdot \sqrt{n}) : & \Theta(n / \lg n) : & O(n / \lg n) : \end{array}$$

2.- (2 pt) Osokoen $A(1..n)$ bektore bat izanik, zenbaki berdinez osaturiko segmentu luzeena mugatuko duen algoritmoa idaztea eskatzen zaizu *zatitu eta irabazi teknika* erabiliaz. Soluzioaren denbora ordena kalkulatu. Eraginkorra dela uste duzu? Arrazoitu zure erantzuna.

3.- (2 pt) Demagun $G(N, A)$ grafo zuzenduak N erpin, A arku eta DEBEKATUA etiketa duten zenbait **erpin** dituela.

BideaDago($G(N, A)$, Hasi, Buka) algoritmoa idatz ezazu, *Hasi* erpinetik *Buka* erpinera bidea badago DEBEKATUA erpinak zeharkatu gabe True itzul dezan, eta False bestela. Algoritmoaren denbora ordena aztertu.

4.- (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 1 euro gutxiago kostatuko zaigu eta hirugarren unitatea, aldiz, 2 euro gutxiago.

Erosketaren erabilgarritasuna erositako produktu bakoitzaren erabilgarritasunaren batura dela jakinik, *Programazio Dinamikoaren* teknika aplikatuz algoritmo bat idatz ezazu gehienez K euro azokako produktuetan gastatuz lor dezakegun erabilgarritasun maximoa mugatzeko.

Honako PISTA erabil dezakezu:

MerkaAzoka (E, i) = lor dezakegun erabilgarritasun maximoa E euro gastatuz soilik $[1...i]$ produktuetan (eta gehienez produktu bakoitzeko 3 unitate erosiz).

5.- (2 pt) n depositu ditugu, i depositu bakoitzaren edukiera d_i litro ($1 \leq i \leq n$) izanik. Depositu hauetako eduki maximoa garraiatu nahi dugu, ezer galdu gabe, K edukiera duen kupela batean erabiliz, gainera, badakigularik $\sum_{i=1}^n d_i > K$. Bestalde, behin depositu bat zabaltzen denean, depositua osorik kupelan hustu behar da, ontzia bota behar delarik. *Backtrack* teknika erabiliz zeintzuk edalontzi komeni den hustea erabaki ezazu.

Algoritmoaren idazketa errazteko, lehendabizi irudika ezazu esplorazio zuhaitza, adarketak aukerak, aukera onargarriak, kimak eta hostoak identifikatzeko. Behar dituzun aldagaien zehaztapena idatzi eta irudikatutako zuhaitzari dagokion kodea egizu. Ez ahaztu aldagaien hasieratzea, lehenengo deia barne.

DEA-II IRAILEKO DEIALDIA (2010-ix-1) - E taldea

SOLUZIO BAT

1.- (2 pt) Ondoren datorkizun $\text{BatuLaginak}(V, a, b)$ algoritmoak $V[a..b]$ bektoreko zenbait osagaien batura kalkulatzeko du. Algoritmoaren exekuzio ordena azter ezazu.

```
function BatuLaginak(V, a, b) return Zenbakia;
if b-a+1 ≤ 3 then return V[(a+b) div 2];
else y := (3*a+b) div 4;
      z := ((a+3b) div 4)+1;
return V[(a+b) div 2] + BatuLaginak(V, a, y) + BatuLaginak(V, z, b);
end if;
end BatuLaginak;
```

Aholkua: algoritmoaren funtzionamendua aztertze adibideren bat erabiltzen baduzu, 30 edo tamaina handiagoko sarrerak erabil itzazu.

Lortu duzun denbora-kostuaren funtzioa ondorengo multzo bakoitzeko elementua den ala ez adieraz ezazu:

$$\begin{array}{lll} O(\lg n) : & o(n) : & \Theta(\sqrt{n}) : \\ \Theta(\lg n \cdot \sqrt{n}) : & \Theta(n / \lg n) : & O(n / \lg n) : \end{array}$$

SOL:

$$\begin{array}{ll} T(n) = 1 & n \leq 3 \\ T(n) = 2 T(n/4) + 1 & n > 3 \end{array}$$

Aldagai aldaketa eta karakteristikoaren metodoa erabiliz: $n=4^i$; $n=4^i = (2^i)^2$; $\sqrt{n} = 2^i$

$$T_k - 2T_{k-1} = 1$$

$$(x-2)(x-1)=0$$

$$T_k = a 2^i + b 1^i = a \sqrt{n} + b$$

Ondorioz:

$$\begin{array}{lll} O(\lg n) : \text{EZ} & o(n) : \text{BAI} & \Theta(\sqrt{n}) : \text{BAI} \\ \Theta(\lg n \cdot \sqrt{n}) : \text{EZ} & \Theta(n / \lg n) : \text{EZ} & O(n / \lg n) : \text{BAI} \end{array}$$

Hedapen metodoa erabiliz, ordena bera lortu genezake:

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

2.- (2 pt) Osokoen $A(1..n)$ bektore bat izanik, zenbaki berdinez osaturiko segmentu luzeena mugatuko duen algoritmoa idaztea eskatzen zaizu *zatitu eta irabazi teknika* erabiliaz. Soluzioaren denbora ordena kalkulatu. Eraginkorra dela uste duzu? Arrazoitu zure erantzuna.

Procedure SL ($A(h,b)$, SL-h, SL-b, SL-l) is

$e := (h+b) \text{ div } 2$;

begin

SL-h:= h; SL-b:= b; SL-l:=1; -- denak desberdinak balira edo length=1

if ($A' \text{length} \leq 2$) then

 If ($A' \text{length} = 2$) and ($(A(h)=A(b))$) then begin SL-h:= h; SL-b:= b; SL-l:=2; end;

 end if;

else begin SL($A(h, e)$, h1,b1,l1);

 SL($A(e+1, b)$, h2,b2,l2);

 TARTEAN($A,e,h3,b3,l3$);

 if ($l3 = \max(l1,l2,l3)$) then begin SL-h:=h3; SL-b:=b3; SL-l:=l3; end;

 elsif ($l1 \leq l2$) then begin SL-h:=h2; SL-b:=b2; SL-l:=l2; end;

 else begin SL-h:=h1; SL-b:=b1; SL-l:=l1; end;

 end if;

end;

3.- (2 pt) Demagun $G(N,A)$ grafo zuzenduak N erpin, A arku eta DEBEKATUA etiketa duten zenbait **erpin** dituela.

BideaDago($G(N,A)$,Hasi,Buka) algoritmoa idatz ezazu, *Hasi* erpinetik *Buka* erpineraino bidea badago DEBEKATUA erpinak zeharkatu gabe True itzul dezan, eta False bestela. Algoritmoaren denbora ordena aztertu.

SOL: Enuntziatua Erromaraino bidearen parekoa da murriztapen gehiago izanik, DEBEKATUrik dauden erpinak ezin baitira zeharkatu. Hau honela, grafoan abiapuntu batek baino gehiago egon litezke: gure HASI grafoko erpin bat da, ez du zertan abiapuntu bat izan behar ($\text{indegree}(\text{HASI})=0$), baina gure abiapuntua izango da. Jomuga betek baino gehiago egon litezke, hots, $\text{Outdegree}(J)=0$, bat BUKA izan liteke, edo ez, BUKA “barruko” erpin bat izan liteke eta. Beraz, (HASI,BUKA) erpin bikote artean eta soilik erpin onargarriak ($\text{not}(\text{DEBEKATUA})$) zeharkatuz biderik dagoen mugatu behar dugu.

Grafoa korritu prozedurak hasieratzeak eta lehenengo deia HASI erpinetik egiteko balio digu eta BUKA erpina finkatzeko. Bidearik (J) j eta BUKA erpinen artean bidea dagoen ala ez jasoko du, behin J guztiz aztertua izan ostean. Egin beharko genukeen deia honako da: GrafoaKorritu($H,B, \text{ExistizenDu}$).

```
procedure GrafoaKorritu ( $G$ : in Grafoa; Hasi, Buka: in Erpina
                        BideaDago: out Boolean) is
    Bisitak, Biderik: Taula(1..N):= (Others=> False);
begin
    Biderik(Buka):=True; Bisitak(Buka):=True;
    -- Biderik(Hasi):=False;
    SK( $G$ , Hasi);
    BideaDago:= Biderik(Hasi)
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);
        HondarraL (A_AuzokideenKopia);
        if not(Bisitak(Lehena)) AND
            (Etiketa( $G$ ,Lehena)/=DEBEKATUA) then
            Bisitak(Lehena):=True; -- Biderik(Lehena):=False
            S_K( $G$ , Lehena);
        end if;
        Biderik(A):= Biderik(A)OR Biderik (Lehena);
        -- Lehena hosto bada, Bukara ez bada iristen,
        -- Biderik(Lehena)=False
        -- Lehena=Buka bada, Biderik(Lehena)=True da
    end loop;
end S_K;
```

Analisia: n erpin dituzten bi taulen hasieratzeak $\Theta(n)$ du. Gehienez arku bakoitza behin tratatuko da eta horren jomugaren erpinaren egoera kontsulta araztea ($\Theta(1)$) eta arkuaren Biderik taulako posizio baten eguneratzea eragingo du ($\Theta(1)$). Ondorioz sakonerako korritzeari gehitutako aginduek ez dute sakonerako korritzearen denbora-ordena handituko, hau $O(n+a)$ izanik.

$$\begin{aligned}
\text{MerkaAzoka}(d,i) &= \text{MerkaAzoka}(d,i-1) && \text{if } d < p_i \\
\\
&= \max\{ \text{MerkaAzoka}(d,i-1), \\
&\quad e_i + \text{MerkaAzoka}(d-p_i, i-1) \} && \text{if } p_i \leq d < 2p_i \\
\\
&= \max\{ \text{MerkaAzoka}(d,i-1), \\
&\quad e_i + \text{MerkaAzoka}(d-p_i, i-1) \\
&\quad 2e_i + \text{MerkaAzoka}(d-2p_i-1, i-1) \} && \text{if } 2p_i \leq d < 3p_i \\
\\
&= \max\{ \text{MerkaAzoka}(d,i-1), \\
&\quad e_i + \text{MerkaAzoka}(d-p_i, i-1) \\
&\quad 2e_i + \text{MerkaAzoka}(d-2p_i-1, i-1) \\
&\quad 3e_i + \text{MerkaAzoka}(d-3p_i-2, i-1) \} && \text{if } 3p_i < d
\end{aligned}$$

5.- (2 pt) n depositu ditugu, i depositu bakoitzaren edukiera d_i litro ($1 \leq i \leq n$) izanik. Depositu hauetako eduki maximoa garraiatu nahi dugu, ezer galdu gabe, K edukiera duen kupela batean erabiliz, gainera, badakigularik $\sum_{i=1}^n d_i > K$. Bestalde, behin depositu bat zabaltzen denean, depositua osorik kupelan hustu behar da, ontzia bota behar delarik. *Backtrack* teknika erabiliz zeintzuk edalontzi komeni den hustea erabaki ezazu.

Algoritmoaren idazketa errazteko, lehendabizi irudika ezazu esplorazio zuhaitza, adarketak aukerak, aukera onargarriak, kimak eta hostoak identifikatzeko. Behar dituzun aldagaien zehaztapena idatzi eta irudikatutako zuhaitzari dagokion kodea egizu. Ez ahaztu aldagaien hasieratzea, lehenengo deia barne.