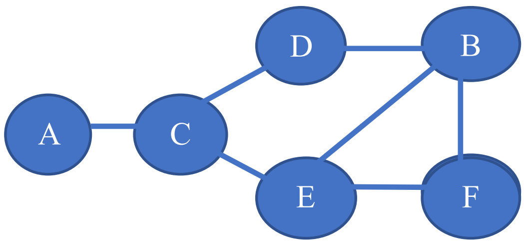
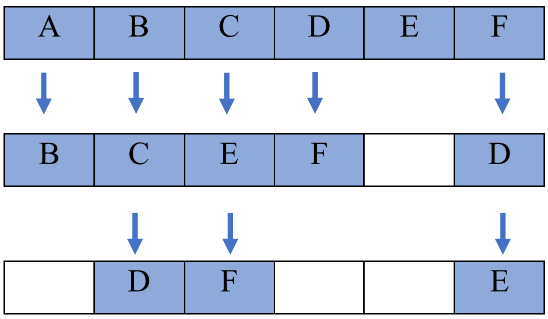
Greining reiknirita – Heimadæmi 5

Dæmi 1

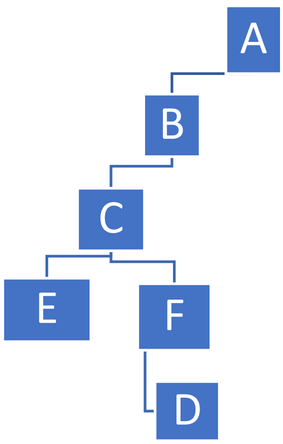
a) Teiknum óstefnt net fyrir fylki:



b) Gerum grennslalista fyrir net:

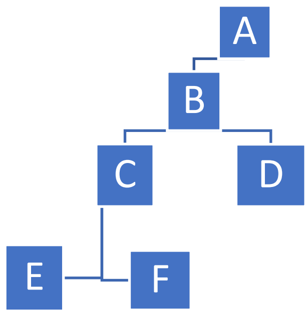


c) Framkvæmum djúpleit á netinu í b-lið:



Hérna erum við að fara frá A í B í C í E, síðan til baka í C í F í D sem gerir röð A,B,C,E,F,D.

d) Framkvæmum breiddarleit á netinu í b-lið:



Hérna erum við að fara í A í B í C til baka í B í D til baka í B í C í E til baka í C í F. Eða röðin er A,B,C,D,E,F.

Dæmi 2

a) Framkvæmum djúpleit á netinu í stafrófsröð.

	PRE:	POST:
A	0	7
B	1	6
C	2	3
D	8	13
E	9	12
F	4	5
G	14	15

	Leggir	Tré	Fram	Bak	Kross
1	A -> B	X			
2	B -> C	X			
3	B -> F	X			
4	D -> A				X
5	D -> E	X			
6	E -> B			X	
7	E -> C			X	
8	E -> H	X			
9	H -> F				X
10	G -> E				X

b) Getum ekki sagt með 100% vissu út frá post gildum, þó að post(x) < post(y) bendir til þess að y hafi verið áfram í leit þegar x var fullunninn og gæti því verið á leið frá y til x. Þetta gefur til kynna að y og x séu tengdir og y var heimsóttur áður en x var fullunninn. EKKI HÆGT.

Dæmi 3

i) <u>Undirverkefnum í orðum</u>: Skilgreinum $F(i, j)$ sem er minnsti fjöldi aðgerða sem þarf til að breyta $A[1 \ldots i]$ í $B[1 \ldots j]$.

ii) <u>Rakningarvenslum og rökstyðja hvernig þau eru fengin (hafið í huga ákvörðun/ágiskun sem tekin er í hverju skrefi):</u> Ef við erum með $A[i] = B[j]$ þá kostar það ekkert og þurfum ekki að breyta neinu. Annars breytum við $A[i]$ í $B[j]$ með því að nota innsetningu, eyðingu og víxlun. Innsetning framkvæmum í tveimur skrefum, setjum $B[j]$ fyrir aftan $A[i]$og nýja táknið. $B[j]$ síðan fjarlægt (ein aðgerð). Með eyðingu fjarlægjum við stak $A[i]$ (ein aðgerð). Og með víxlun þá víxlum við á $A[i]$ og $A[i-1]$ þar sem $B[j] = A[i-1]$ og $A[i] = B[j-1]$ (0 aðgerð). Rakningarvensl eru því:

$$F(i, j) = F(i-1, j-1) \; ef \; A[i] = B[j]$$
$$F(i, j) = F(i-2, j-2) \; ef \; A[i] = B[j-1] \; og \; B[j] = A[i-1]$$
$$Annars \; 1 + \min\left(F(i-1, j), F(i, j-1)\right)$$

iii) <u>Röð undirverkefna:</u> Sjáum að rakningarvenslunum að F(i, j) eru háð minni undirverkefnum.

iv) <u>Grunntilviki:</u>
1. $F(0, j) = j$ (innsetningar þegar A er tómur)
2. $F(i, 0) = i$ (eyðingar þegar B er tómur)

v) <u>Hvernig lausn á upphaflegu verkefni er fengin:</u> Þegar þetta er búið mun $F(n, m)$ gefa okkur lausn.

vi) <u>Hvernig hægt er að koma í veg fyrir endurtekna útreikninga með því að geyma milliniðurstöður í viðeigandi gagnagrind:</u> Komum í veg fyrir endurtekinn útreikning með því að geyma niðurstöður í memoization-töflu og vísa í geymdu gildin í stað þess að endureikna þau.

vii) <u>Tímaflækju og rökstyðja hvernig hún er fengin:</u> Tímaflækjan verður $O(n * m)$ þar sem við erum að erum að sækja gildi í töflu.

Dæmi 4

i) <u>Undirverkefnum í orðum</u>: Skilgreinum $SUM(v)$ sem er hæsta summa í hluttré með rót v.

ii) <u>Rakningarvenslum og rökstyðja hvernig þau eru fengin (hafið í huga ákvörðun/ágiskun sem tekin er í hverju skrefi):</u> Þurfum að ákvarða hvort hnútur með rót í x sé partur af hæstu summu eða ekki, gerum það með rakningarvenslunum.

$$SUM(x, 1) = \max\left(W_v + SUM(X_v, j+1), SUM(X_h, k+1)\right)$$

Hérna er $X_v = vinstra \; barn$ og $X_h = hægra \; barn$, j er fjöldi hnúta á X_v og k er fjöldi hnúta á X_h, Þetta er prófað á að summa á öll börn x á meðan hnútafjöldi er undir 1.

iii) Röð undirverkefna: Förum frá laufum í rót, þ.e. upp tré.

iv) Grunntilviki:
1. $SUM(v) = 1\ ef\ W_v\ = 1$ og f(v) = 1 og v er lauf í T

v) Hvernig lausn á upphaflegu verkefni er fengin: Fáum hana með $SUM(r)$

vi) Hvernig hægt er að koma í veg fyrir endurtekna útreikninga með því að geyma milliniðurstöður í viðeigandi gagnagrind: Komum í veg fyrir endurtekinn útreikning með því að geyma niðurstöður í töflu.

vii) Tímaflækju og rökstyðja hvernig hún er fengin: Tímaflækjan er $O(n)$ það er því það er n hnútar í tré og förum einusinni í hann því við geymum gildi í töflu. Hver hnútur tekur fastan tíma.

Dæmi 5

i) Undirverkefnum í orðum: Skilgreinum tvívíðan lista $F(i, j)$ þar sem i eru fyrstu stökin í A og j stökin sem er valinn af þeim fjölda. $F(i, j)$ skilar TRUE ef hægt er að velja j stök af i stökum úr A þannig að það sé tenging á milli para j en ekki $j + 1$ upp í i.

ii) Rakningarvenslum og rökstyðja hvernig þau eru fengin (hafið í huga ákvörðun/ágiskun sem tekin er í hverju skrefi): Skoðum stak i og leitum að tengdu pari í vensluðu stökunum. Ef stak er ekki með tengingu eða hefur tengt par sem getur verið með þá notum við rakningarvenslin:

$$F(i, j) = F(i - 1, j - 1)\ ef\ par\ finnst$$
$$F(i, j) = F(i - 1, j)\ ef\ par\ finnst\ ekki$$

Annars verður $F(i, j) = False$

iii) Röð undirverkefna: Fyllum upp í F frá $F(0,0)$ upp í $F(n, k)$.

iv) Grunntilviki:
1. $F(0,0) = TRUE$ (alltaf hægt að velja 0 stök af 0 mögulegum).
2. $F(i, 0) = TRUE$ fyrir öll i (Alltaf hægt að velja 0 stöku sama hvað er í pott).
3. $F(0, j) = FALSE$ fyrir öll $j\ > 0$ (Ekki alltaf hægt að velja fleiri en 0 stök af 0 mögulegum).

v) Hvernig lausn á upphaflegu verkefni er fengin: Fáum að $F(n, k)$

vi) Hvernig hægt er að koma í veg fyrir endurtekna útreikninga með því að geyma milliniðurstöður í viðeigandi gagnagrind: Komum í veg fyrir endurtekinn útreikning með því að geyma niðurstöður í töflu.

vii) Tímaflækju og rökstyðja hvernig hún er fengin: Tímaflækjan er $O(n * k)$ eða margliðu tímaflækja, það er því við höfum n val fyrir i frá 0 til n og sama á við k.