

1. feladat (10 pont)

Tömörítse **Huffman algoritmussal** az alábbi szöveget. Építse fel a **kódfát**, határozza meg az **ábécé betűinek kódját**. Adja meg naiv tömörítéssel és a Huffman kódfa alapján kapott **tömörített hossz** bitekben. (*Naiv tömörítés esetén a jel hosszának azt a legkisebb bitszámot kell tekinteni, amin az ábécé betűi megkülönböztethetők.*)

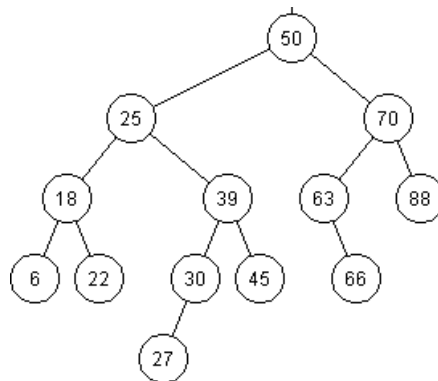
KEREK_KERTEMBEN_BERREG

2. feladat (9 + 5 pont)

(a) A felsorolt kulcsok egymás utáni beszúrásával építsen a következő számsorozatból **AVL fát**. Feltétlenül jelezze **minden forgatás előtt** a fa csúspontjainak egyensúlyát, jelölje a transzformációban résztvevő részfat, és adja meg a forgatás típusát.

66 82 33 74 16 48 37 55 60 50 58 20

(b) Címkezza meg az alábbi AVL fa csúcsait egyensúlyuknak megfelelően. **Törölje a 70-es** kulcsot a tanult algoritmus szerint, majd mutassa be a fa helyreállításának lépéseit.



3. feladat (10 pont)

Adott a következő **B+ fa** ($d=4$, azaz 4 pointer található minden csúcsban). (a) **Rajzolja le**, hogyan néz ki a fa, majd (b) **szűrje be** a fába a **41** és **26** kulcsokat (mindig az **eredeti** fába szűrjön be). (c) Az eredeti fából **törölje** a **3** kulcsot.

Rajzolja le a műveletek utáni állapotot – elég a fa gyökerét és a megváltozott részfat lerajzolni!

{ [(3 5) 10 (12 14 15) 22 (22 25 28) 30 (31 33)] 40 [(42 43 44) 46 (46 48) 50 (50 52)] }

4. feladat (12 pont)

Adott egy láncoltan ábrázolt **általános fa**. Egy csúcsban az első gyerekekre és testvérré mutató pointerok vannak. Készítse el a következő kereső algoritmust: megadunk egy kulcsot, **Írjuk ki** azoknak a **szülőknak a címét**, amelyek gyerekei között szerepel a megadott kulcs. Ha nem található az adott kulcs a fában, akkor kiírja, hogy „nem található a kulcs”. Elképzelhető, hogy az adott kulcs több helyen is szerepel a fában, akkor **mindegyik** előforduláshoz írja ki a szülőt!

5. feladat (14 pont)

Adott egy **irányított élsúlyozott gráf éllistas** reprezentációja az Adj[1..n] tömb segítségével. Az éllistas csúcs szerint *növekvően* rendezettek. Adj tömb az éllistas első elemére mutató pointerokat vagy NULL értéket tartalmaz. Írjon hatékony algoritmust, amely törli egy adott x csúcsba mutató valamennyi élt. (Párhuzamos élek előfordulhatnak a gráfban, ha élsúlyuk különböző. Például: (3,5) 4 súlyú és (3,5) 10 súlyú él létezhet.)