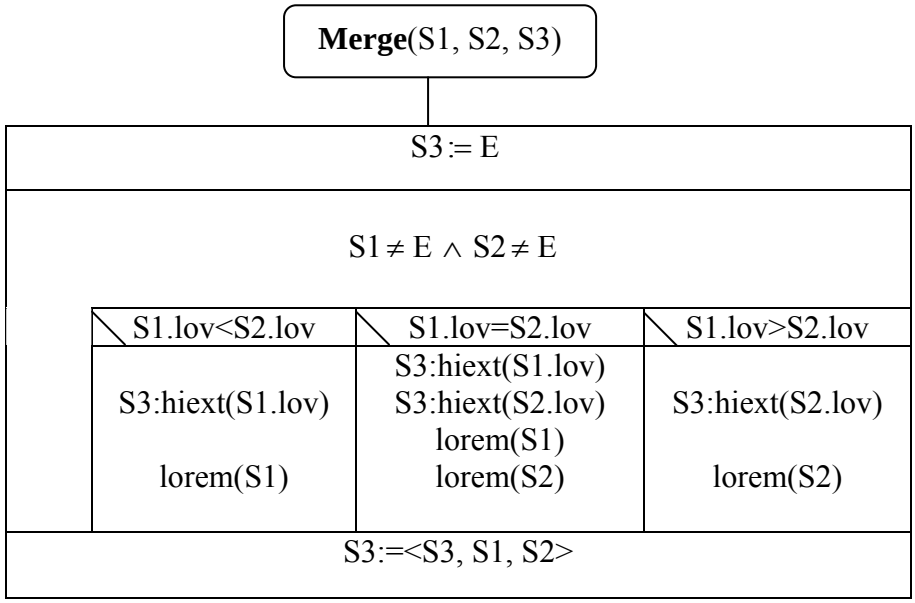
13. Összefuttatásos rendezés (Merge Sort)

A merge-sort alapját két rendezett sorozat összefuttatása képezi.
Ha S1 és S2 két rendezett sorozat, akkor az alábbi összefuttatás után S3 rendezett lesz.
Az algoritmus a *Bevezetés a programozásba* című tárgyból jól ismert 2-1 típusú elemenkénti feldolgozás egy változata. Eltérés: ha két azonos értéket olvasunk S1-ből, illetve S2-ből, akkor mindkettőt ki kell írni S3-ba, hiszen a rendezés során elemet nem veszíthetünk.



Továbbá ebben a változatban a CF-ben „∧” áll, azaz addig megy a ciklus, amíg valamelyik sorozat el nem fogy.

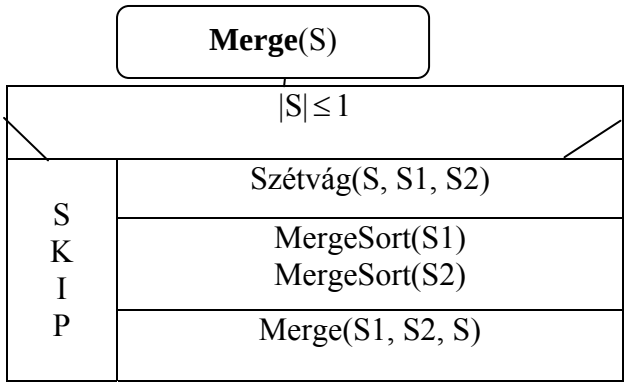
← Konkatenáció; legalább az egyik sorozat üres

Az összefésülést megszokták fogalmazni a következő adatszerkezetre:

- láncolt listákra – ugyanezzel a CF-fel; pointerek kezelésével;
- szekvenciális file-okra – a CF-ben „∨”-gyal; sx, dx, x: read-del;
- tömbökre – ekkor az elágazás feltételei bonyolultabbak (jól ismert);

(Feltételezzük, hogy ezeket a változatokat bárki le tudja írni.)

A merge-sort rekurzív algoritmusa ezek után a következő:



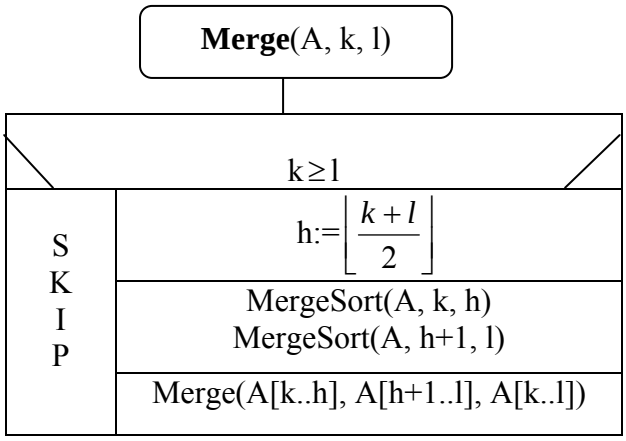
Az üres vagy 1 elemű sorozat rendezett.

S-et két részre vágjuk.

A két részt külön-külön rendezzük ugyanezzel az eljárással.
A két rendezett részsorozatot összefuttatjuk: S rendezett lesz.

A Szétvág(S1, S2, S3) eljárás általában két körülbelül azonos hosszú részre vágja S-et, így várható az optimális működés, azaz a legkevesebb összehasonlítás.

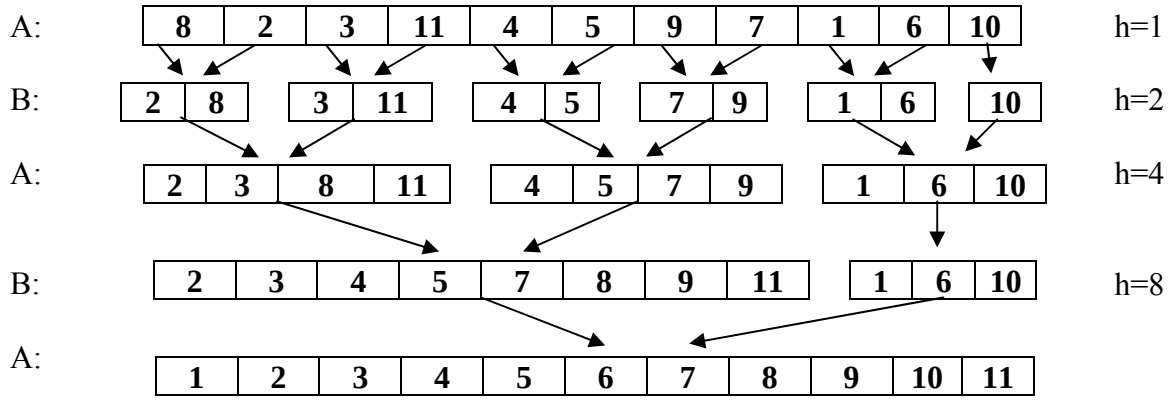
A rendezést is ugyanarra a három adatszerkezetre szokták leírni, mint az összefuttatást.(Tömbök esetén nem foglalkoznak azzal, hogy hogyan lehet O(1) segédmemóriával megoldani a rendezést, hanem egy segédtömbre futtatnak össze, és annak visszamásolják az eredményét az eredetibe.) Tömbökre így néz ki:



Külső hívás:
MergeSort(A, 1, n)

(Ehhez kell egy B segédtömb.)

Tömbökre szokás egy másik, iteratív stratégiát követni.
Minden lépésben végigmegyünk a tömbön az egyre növekvő $h = 1, 2, 2^2, 2^3, ..., 2^{\lceil \log_2 n \rceil - 1}$ lépésközzel, és az ilyen hosszú szomszédos tömbrészleteket fésüljük össze. Az A tömbből a B tömbbe futtatunk össze, azután a B-ből az A-ba, és így tovább felváltva. Lehet, hogy valamely iterációban az utolsó pár második tagja rövidebb, mint h; és az is lehet, hogy páratlan sok tömbrészlet van, ilyenkor az utolsót csak átmásoljuk.
(Ha végül a B tömbben végződne az összefésülés, akkor még visszamásoljuk az eredményt az A-ba.)
Az algoritmust nem írjuk le, csak példával illusztráljuk a működését:



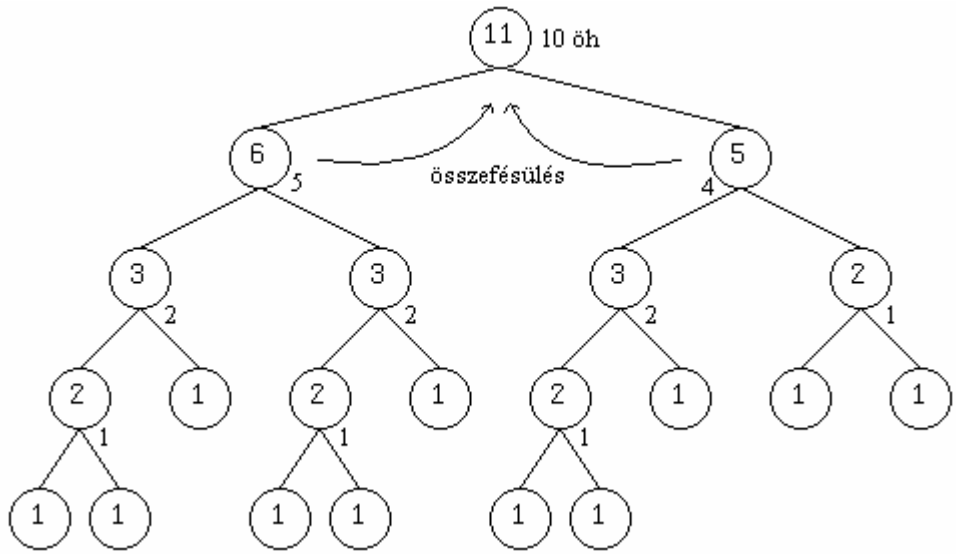
Hatékonyság elemzés

A merge-sort-nál is az összehasonlítások számát becsüljük (noha a mozgások számát könnyebben, pontosabban lehetne), és csak a legrosszabb esetet vizsgáljuk.

A Merge eljárás egy n_1 és egy n_2 hosszú sorozat összefésülésekor a legrosszabb esetben $n_1 + n_2 - 1$ összehasonlítást végez: minden összehasonlítás után 1 elem kerül kiírásra, ha a két elem nem egyenlő; ha utoljára még mindkét sorozatból maradt 1-1 elem, akkor valóban a legtöbb összehasonlítást idéztünk elő, viszont az utolsó összehasonlítás után már mindkét elemet kiírjuk (és a két sorozat kiürül).
Tehát: $MÖ_{Merge}(n_1, n_2) = n_1 + n_2 - 1$.

Ezt úgy is lehet írni , hogy: $MÖ_{Merge}(n) = n - 1$, ha n a két sorozat együttes hossza.

A MergeSort eljárás elemzésénél a közel egyenlő szétvágást vesszünk alapul. Nézzük meg például egy n=11 hosszú sorozat rendezése esetén azt, hogy milyen hosszú részekre vágja az eljárás a (rész)sorozatokat a rekurzív hívások egyes szintjein. Az alábbi lineáris fa csúcsaiban a hosszak szerepelnek. Az egyes csúcsok mellett szereplő szám azt mondja meg, hogy az illető részsorozat hány összehasonlítással kapható meg, két rendezett rész összefésülésével. Az összehasonlítások maximális száma a példában: 10+(5+4)+(2+2+2+1)+(1+1+1)=29.



Látható, hogy az egyenlő szétvágások bináris fája majdnem teljes. Ez általában is igaz.
A fa magassága $4 = \lceil \log_2 11 \rceil$;

általában $h = \lceil \log_2 n \rceil$.

A fának a leveleihez nem tartozik összehasonlítás-szám (0), tehát éppen h szinten kell összegezni az összehasonlítások számát. Ezek szintenkénti összege egyre csökken, de minden esetre felülről becsülhető $(n-1)$-gyel. Így:

$$MÖ_{MS}(n) \le (n-1)\lceil \log_2 n \rceil = O(n \log_2 n).$$

Ha tehát azt szeretnénk, hogy $n = 2$ esetén is igaz legyen a képlet, akkor nem is javítható (könnyen, ilyen módon) a fenti képet.