A következő címről ötletturi eredménye

<http://www.cs.bme.hu/sza/anim.html>

Előre mondom, hogy mindent leírtam ami eszembe jutott, válogatás nélkül, tekintet nélkül a megvalósíthatóságra vagy annak nehézségére. Az egy külön lépcső, hogy eldöntsük, mi kell és mi nem, illetve mik a prioritások.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Beszúrásos rendezés |  |  | [Eredeti](http://al.ei.tuat.ac.jp/%7Esekisita/isort-e.html) | [SZIT mirror](http://www.cs.bme.hu/%7Egsala/alg_anims/3/isort-e.html) |
| * Számok reprezentálása különböző hosszúságú oszlopokkal. (Kevésbé érzem fontosnak, állítható lehetne) * **Elkülöníteni összehasonlításnál, azt amit hasonlítunk, amivel.** (Indexelt tömb elemnél lenne érdekes) * Értékadás **után vastagon maradjanak a változások**, pl cserénél, de a következő utasítással tűnjenek el. * Valahogy jó lenne (Pl változó beállítások varázslóban, engedni, hogy egy változót indexnek jelöljünk meg és akkor ezt rendelni valahogy az ábrához. Esetleg megadható lenne, hogy melyik tömb lenne indexelve vele) * Ha indexelt változót hasonlítunk, valami mással, konstans, vagy másik változó, ami nem index, azt is jó lenne jelölni. Általánosságban, egyik szín mit hasonlítasz ( ball operandus ) másik szín mivel hasonlítod ( jobb operandus ) konjugció,vagy diszjunkciónál pedig akár lehetne több ilyen pár is. |  |  |  |  |
| Buborék rendezés |  |  | [Eredeti](http://al.ei.tuat.ac.jp/%7Esekisita/bsort-e.html) | [SZIT mirror](http://www.cs.bme.hu/%7Egsala/alg_anims/3/bsort-e.html) |
| Kupac rendezés   * Fára szintén elmondható, hogy ami jó lenne látszódjon, mit mivel hasonlítunk, majd **a változások legyenek láthatók a következő lépésig. ( Mi egy következő lépés ? StepIN ,Step, Step over esetén esetleg állítható lehetne. Megjelenítés leíróba, hogy a jelölések meddig éljenek utasításszám, step egyéb alapján. )** * A mit lássunk, mit ne lássunk, mibe lépjünk bele, mibe ne, fa mellé kell egy, az ebből a fából létrehozott ( a kipipált elemekből ) egy részletesebb tulajdonság beállító. Változónként. * **Megjelenítési módok váltása**. Fa van a kódban, de a megjelenítésnél tömb legye és fordítva is. Menet közben és a leíróban is állítható legyen. * A Színezésnél konzisztens jelölés, mindenhol ugyanaz legyen a balérték színe és a jobb érték színe. Esetleg készletek felüldefiniálhatók lehessenek az user által. * **Értékadásnál, minek adsz értéket egyik szín, mit adsz értékül másik szín** HA a jobb oldalon kifejezés áll, meggondolandó lenne, a kifejezés egyes tagjait ugyanolyan színnel jelölni, jelezve, hogy ezen értékek felhasználásával történik, az értékadás. Konstans temporális értékek megjelenítése ? Kikapcsolható formában csakis. * Jó **lenne változótulajdonságnak megadni, hogy milyen változó** **az. Pl : kimenet**. PL egy lista, amibe gyűjtöd az adott algoritmus eredményét. Azt esetleg máshogy, máshova jelenítené, meg mert az nem annyira érdekes, csak mondjuk abban a pillanatban, amikor belekerül egy elem. Ez a tulajdonságlista, változónként egy bővíthető dolognak kéne lennie és elsősorban GUI-n generált formon összekattolható. ( De generáljunk belőle XML –t oszt lehet azt is editálgatni, ha valaki erre gerjed …. ) * FV –ek bemenő paraméterei is konfigurálhatóak legyenek. Pl legjobb a Quick sort. Lehessen olyat, hogy fv bemenő paramétere tömb, a,b,c Tömbből mutasd az a,b intervallumot csak (A többit esetleg nyilacskával ki-be csukogathatóra csinálni, vagy egyáltalán nem megjeleníteni, vagy átlátszósítani vagy … vagy ☺) Ez a helyben rendező algoritmusoknál, mint pl QS jó lenne, mert ott mindig cska a szelet érdekel minket. A jelölés olyan legyen, hogy egyértelmű legyen, hogy az nem csak egy rövid tömb, hanem egy egész tömb, minket érdeklő része. * A már sokat emlegetett változótulajdonságok, fv tulajdonságoknál van még egy fontos szempont. Módosított program esetén ebből mi maradjon meg, mi nem. EZ sok tökölés lesz, már agyalok rajta, mert már most előjött a ami látszódjon, mi nem problémánál. Itt azért minél okosabbra jó lenne, hogy minél több beállítás megmaradjon. Amit a résztömb megjelenítéséről mondtam, az valamiféle adatkötés szerű dolognak kéne lennie és elég rugalmasnak. |  |  | [Eredeti](http://www.cs.auckland.ac.nz/software/AlgAnim/heapsort.html) |  |
| Quicksort   * Arra jutottam, hogy a legjobb talán az lenne, ha lehetne állítani, hgoy egy változó (Összetett adatszerkezet) Mely részeit lássuk, ne lássuk, lásuk halványan, rejtsük el nyilacskával. Ez szépen általánosan megfogalmazva fedi, a fák sorba való betételét (Nekem így jobban tetszik ahogy van, mivel én így képzelem el ..), tömbök részeinek kiemelését, pl QS-nél. Stb. |  |  | [Eredeti](http://www.cs.auckland.ac.nz/software/AlgAnim/qsort.html) |  |
| **Rendezőalgoritmusok (**Beszúrásos,  quick,  buborék, kupacos)   * Hatékonyság. Összehasonlítások, cserék, értékadások száma. Valami szép grafikonnal. Itt lehetne olyat, hogy ahol az összehasonlítás van, abba na ciklusban, az összehasonlításokhoz képest hány csere volt. Ciklusra valahogy tudni kell hivatkozni ( Ezt lehet a fával, amit épp gyártok ) illetve valami szép grafikonon, hgoy iterációnként mennyi volt az összehasonlítás. Statisztika készítés. Futtasd le n –szer és mondj egy statisztikát. (Futási idő becslés, progress bar.) * Hatékonyságnál pl a program valmiféle speciális konfigurációjú szintaxfáját lehetne használni arra, hgoy miket számoljon és esetleg milyen counterbe ! (Vízió : jobb klikk egy expressionra , add counter, name stb.) * Megjelenítés előre konfigolása. Elrendezés. Jó lenne pl, ha megadhatnám, hogy mit, hol és mekkorában akarok látni. Ehhez négyzeteket rajzolnék a megjelenítési területre és hozzárendelnék egy változót. PL. Hangsúlyozom, ötletek . ☺ * Default counterek. Amik, mindent számolnak. Ki-be kapcsolható. Hatékonyság miatt. * Változók kötése. Pl változó megjelölése, hogy ennek a tömbnek (akár többnek is) az indexe. Ekkor a változók nevének megjelenítése az adott index felett. **Tömbindexek kiírása !** Ez nagyon kéne. |  |  | [Eredeti](http://www.cs.smith.edu/%7Ethiebaut/java/sort/) | [SZIT mirror](http://www.cs.bme.hu/%7Egsala/alg_anims/1/demo.html) |
| Rendezőalgoritmusok (Beszúrás, **quick!**, **összefésülés!**, kupac)   * Ha már perverzkedünk akkor adjunk neki. Szimultán több algoritmus futtatása. 2 ablak 2 algoritmus ugyanarra bemenetre, lássuk, melyik kevesebb lépés, lássuk, melyik, hol mit csinál. Ahol ennek értelme van. Ez már többablakos, és inkább több monitoros, vagy egy monitoron, de akkor nagyon szűkre szabva, hogy mi jelenjen meg. Ez ugye állítható. |  |  | [Eredeti](http://www.cs.pitt.edu/%7Ekirk/cs1501/animations/Sort1.html) |  |
| **Rendezőalgoritmusok összehasonlítása** |  |  | [Eredeti](http://www.cs.brockport.edu/cs/java/apps/sorters/race_sorters/sortchoiceinp.html) | [SZIT mirror](http://www.cs.bme.hu/%7Egsala/alg_anims/2/sortchoiceinp.html) |
| **Láda rendezés** |  |  | [Eredeti](http://www.cs.auckland.ac.nz/software/AlgAnim/binsort.html) |  |
| **Radix rendezés**   * Edény rendezés a külön állatfaj. Ide spec megjelenítés, bár egy felül nyitott stack ábrázolás is elég szerintem. |  |  | [Eredeti](http://www.cs.auckland.ac.nz/software/AlgAnim/radixsort.html) |  |
| **Gráfokról egyenlőre nem elmélkednék, kíváncsi vagyok Lovász miket mutat.** |  |  |  |  |
| PERT |  |  | [Eredeti](http://www.renyi.hu/%7Esali/pert.html) |  |
| **Euler-kör** |  |  | [Eredeti](http://www.cs.sunysb.edu/%7Eskiena/combinatorica/animations/euler.html) | [SZIT mirror](http://www.cs.bme.hu/%7Ekiskat/sza/euler.gif) |
| **Hamilton-kör** |  |  | [Eredeti](http://www.cs.sunysb.edu/%7Eskiena/combinatorica/animations/ham.html) | [SZIT mirror](http://www.cs.bme.hu/%7Ekiskat/sza/ham.gif) |
| **Minimális hosszúságú Hamilton-kör** (több féle közelítés) |  |  | [Eredeti](http://www.stockheim.net/touropt.html) |  |
| Kruskal algoritmus |  |  | [Eredeti](http://students.ceid.upatras.gr/%7Epapagel/project/kruskal.htm) |  |
| Dijkstra,  **Kruskal,** Ford-Fulkerson |  |  | [Eredeti](http://www-b2.is.tokushima-u.ac.jp/%7Eikeda/suuri/kruskal/Kruskal.shtml) |  |
| **Szélességi bejárás** |  |  |  | [SZIT mirror](http://www.cs.bme.hu/%7Egsala/alg_anims/3/graph2-e.html) |
| **Mélységi bejárás** |  |  |  | [SZIT mirror](http://www.cs.bme.hu/%7Egsala/alg_anims/3/graph-e.html) |
| Szélességi és mélységikeresés, Dijkstra, Kruskal |  |  | [Eredeti](http://cs.smith.edu/%7Ethiebaut/java/graph/demo.html) |  |
| Dijkstra |  |  | [Eredeti](http://www.cs.auckland.ac.nz/software/AlgAnim/dijkstra.html) |  |
| **Dijkstra**, Kruskal, Ford-Fulkerson |  |  | [Eredeti](http://www-b2.is.tokushima-u.ac.jp/%7Eikeda/suuri/dijkstra/Dijkstra.shtml) |  |
| Dijkstra (Ausztrália térképen) |  |  | [Eredeti](http://www.ifors.ms.unimelb.edu.au/tutorial/dijkstra/island.html) |  |
| Ford-Fulkerson |  |  | [Eredeti](http://riot.ieor.berkeley.edu/%7Eolivier/graal-flow/graal.html) |  |
| Dijkstra, Kruskal, **Ford-Fulkerson** |  |  | [Eredeti](http://www-b2.is.tokushima-u.ac.jp/%7Eikeda/suuri/maxflow/Maxflow.shtml) |  |
| Ford-Fulkerson, Dijkstra, **Bellman-Ford**, DFS |  |  | [Eredeti](http://links.math.rpi.edu/applets/appindex/graphtheory.html) |  |
| Floyd-Warshall algoritmus |  |  | [Eredeti](http://www-m9.ma.tum.de/java-applets/routenplanung/Algorithmen/FloydWarshall.de.html) |  |
| Floyd-Warshall algoritmus |  |  | [Eredeti](http://www.pms.informatik.uni-muenchen.de/lehre/compgeometry/Gosper/shortest_path/shortest_path.html#visualization) |  |
| **Floyd** |  |  | [Eredeti](http://students.ceid.upatras.gr/%7Epapagel/project/kef5_7_2.htm) |  |
| Gráf rajzolás |  |  | [Eredeti](http://www.cs.rpi.edu/projects/pb/graphdraw/index.html) |  |
| **Duális gráf** |  |  | [Eredeti](http://math.ucsd.edu/%7Efan/152/arch/dualplanar/) |  |
| Párosítás páros gráfban |  |  | [Eredeti](http://www.math.sfu.ca/%7Egoddyn/Courseware/Visual_Matching.html) |  |
| Páros gráf élszínezése (párosítás) |  |  | [Eredeti](http://www.cgl.uwaterloo.ca/%7Essiu/school/courses/cs760k/) |  |
| Minimális súlyú párosítás |  |  | [Eredeti](http://www.math.sfu.ca/%7Egoddyn/Courseware/Visual_Matching.html) |  |
| Vegyes, linkek mindeféle animációra |  |  | [Eredeti](http://www.hig.no/%7Ealgmet/animate.html) |  |
| Szomszédossági mátrix |  |  | [Eredeti](http://www.student.seas.gwu.edu/%7Eidsv/idsv.html) |  |
| **Kongruencia egyenlet megoldása** |  |  | [Eredeti](http://www.mathlab.mtu.edu/COURSES/holt/dnt/lincong3.html) |  |
| **Euklideszi algoritmus** |  |  | [Eredeti](http://www.math.sc.edu/%7Esumner/numbertheory/euclidean/euclidean.html) |  |
| **Erasztotenészi szita** |  |  | [Eredeti](http://cs.oregonstate.edu/%7Eminoura/cs261/javaProgs/bitSet/ESieve.html) |  |
| **Faktorizálás** |  |  | [Eredeti](http://www.cryptographic.co.uk/factorise.html) |  |
| **Titkosítások** |  |  | [Eredeti](http://pages.central.edu/emp/LintonT/classes/spring01/cryptographyframeset.htm) |  |
| **RSA titkosítás** |  |  | [Eredeti](http://islab.oregonstate.edu/koc/ece575/02Project/Mor/) |  |
| RSA titkosítás |  |  | [Eredeti](http://cisnet.baruch.cuny.edu/holowczak/classes/9444/rsademo/) |  |
| Turing gép |  |  | [Eredeti](http://www.turing.org.uk/turing/scrapbook/tmjava.html) |  |
| **Csoportok** |  |  | [Eredeti](http://math.ucsd.edu/%7Ejwavrik/Groups15/Groups15.html) |  |
| **Csoportok** |  |  | [Eredeti](http://www.shef.ac.uk/%7Edaj/) |  |

Kivonat: elkezdtem kigyűjteni, összefoglalva, amiket fentebb írtam, de csak elkezdtem. Kb 30% ha ittvan.

**Láthatósági pipafa:**

Változók láthatósága ki/be  
FV -ekbe belépjünk/ne lépjünk

**Változó tulajdonságok:**

-Változó típus megadása: index, segéd, kimenő, temporális, fontos, nem fontos stb  
 -Adatkötés szerűség. A változó megjelenítése függjön B változó értékétől stb.  
 -Változó default megjelenítési formája : fa,tömb,speciális, USerDefined ehhez kell egy jó kis interface a megjelenítéshez.  
-Elsősorban összetett adatszerkezeteknél, mi látszódjon az adatszerkezetből, helyzethez, változóhoz értékéhez kötve. (Helyzet, mint fv, ciklus egyéb vezérlési szerkezetben.) Változó értéke, mint amit QS –nél írtam.  
-Egyes típusok konfigurálása. (Típusonként, változónként) Pl tömböknél akarjuk e látni az indexeket. PriorityQueue –nél, akarjuk e jelölve látni folyamatosan a max, minimális elemet. Stb.

**Függvény tulajdonságok:**

A bemenő paraméterek tulajdonságainak meghatározása. Itt is lehetne olyan adatkötés szerűség.

**Animációk:**

Minimalizálni, mivel zavaró, és figyelem elterelő. Mindenképpen kikapcsolható, ne ez legyen a fő irány, inkább a testreszabhatóság, konfigurálhatóság kényelmes steppelgethetőség. A legtöbb esetben az animáció elsőre szép, de aztán már értelmét veszti. Pontosítok, csökken a hatása exponenciálisan. De ez az én véleményem. ( Szeretem a kis animációkat, emrt jópofák és barátibbá teszik a dolgokat. )