eucgvuts Group

John D. Baker

https://github.com/bakerjd99/jacks/blob/master/eucgvuts/eucgvuts.ijs

SHA-256: 377248abb1dde3712df43035cb4661eee58fc48d3d8a3d3fdec065abd4c56093

June 23, 2023

Contents

eucgvuts Overview

eucgvuts Source Code

```
NB.*eucgvuts t-- various Euclid graphviz digraph utils.
NB.
NB. verbatim: interface word(s):
NB. ------------------------------------------------------------------------------
NB.  eucjoycebkdeps - all justifications from Joyce book html files
NB.  eucjoycecncts  - format Joyce node connections
NB.  eucjoycedeps   - extract noted book dependencies from Joyce html
NB.  eucjoycehtml   - html from David Joyce's online Elements
NB.  eucjoycetabs   - extract dependency tables from Joyce html
NB.  eucsortBgv     - second sort and format euclid book digraphs
NB.  eucsortgv      - sort euclid book digraphs
NB.  gvclustoff     - dot code marked cluster(s) off
NB.  gvcluston      - dot code marked cluster(s) on
NB.
NB. created: 2023jun23
NB. changes: -----------------------------------------------------

coclass 'eucgvuts'
NB.*end-header


NB. dot code off cluster marks
CLUSTOFFMARKS=: <;._1 ' ////---cluster-start ////---cluster-end'

NB. carriage return character
CR=: 13{a.
```

```
NB. interface words (IFACEWORDSeucgvuts) group
IFACEWORDSeucgvuts=: <;._1 ' eucjoycebkdeps eucjoycecncts eucjoycedeps eucjoycehtml eucjoycetabs eucsortBgv
>..> eucsortgv gvclustoff gvcluston'


NB. line feed character
LF=: 10{a.

NB. root words (ROOTWORDSeucgvuts) group
ROOTWORDSeucgvuts=: <;._1 ' IFACEWORDSeucgvuts ROOTWORDSeucgvuts VMDeucgvuts eucgvuts_hashdateurl eucjoyceb
>..>kdeps eucjoycecncts eucjoycedeps eucjoycehtml eucjoycetabs eucsortBgv eucsortgv gvclustoff gvcluston'


NB. 13 Euclids Elements books in Roman numerals
RomanElementsBooks=: <;._1 ' I II III IV V VI VII VIII IX X XI XII XIII'

NB. tab character
TAB=: a.{~9

NB. version, make count and date
VMDeucgvuts=: '0.5.0';7;'23 Jun 2023 13:28:22'

NB. mark end of book dot digraph nodes
eucENDBOOKDEPS=: '//===end-book-deps'

NB. mark end of node attributes
eucENDNODEATTRS=: '//===end=node-attributes'
```

```
NB. mark start of book dot digraph nodes
eucSTARTBOOKDEPS=: '//===start-book-deps'

NB. mark start of node attributes
eucSTARTNODEATTRS=: '//===start-node-attributes'

ncolorDEFINITION=: 'greenyellow'

ncolorNOTION=: 'darksalmon'

ncolorPOSTULATE=: 'lightblue'

ncolorTERMINAL=: 'gold'

NB. retains string after first occurrence of (x)
afterstr=: ] }.~ #@[ + 1&(i.~)@([ E. ])

NB. trims all leading and trailing blanks
alltrim=: ] #~ [: -. [: (*./\. +. */\) ' '&=

NB. trims all leading and trailing white space
allwhitetrim=: ] #~ [: -. [: (*./\. +. *./\) ] e. (9 10 13 32{a.)"_

NB. signal with optional message
assert=: 0 0"_ $ 13!:8^:((0: e. ])`(12"_))
```

```
NB. attribute xml BEGIN and END tags
atags=: '<'&,@,&' ' ; '</'&,@,&'>'

NB. retains string (y) before last occurrence of (x)
beforelaststr=: ] {.~ 1&(i:~)@([ E. ])

NB. retains string before first occurrence of (x)
beforestr=: ] {.~ 1&(i.~)@([ E. ])

betweenstrs=: 4 : 0

NB.*betweenstrs v-- select sublists between  nonnested delimiters
NB. discarding delimiters.
NB.
NB. dyad:  blcl =. (clStart;clEnd) betweenstrs cl
NB.        blnl =. (nlStart;nlEnd) betweenstrs nl
NB.
NB.   ('start';'end') betweenstrs 'start yada yada end boo hoo start ahh end'
NB.
NB.   NB. also applies to numeric delimiters
NB.   (1 1;2 2) betweenstrs 1 1 66 666 2 2 7 87 1 1 0 2 2

's e'=. x
llst=. ((-#s) (|.!.0) s E. y) +. e E. y
mask=. ~:/\ llst
(mask#llst) <;.1 mask#y
)
```

```
changestr=: 4 : 0

NB.*changestr v-- replaces substrings - see long documentation.
NB.
NB. dyad:  clReps changestr cl
NB.
NB.   NB. first character delimits replacements
NB.   '/change/becomes/me/ehh' changestr 'blah blah ...'

pairs=. 2 {."(1) _2 [\ <;._1 x      NB. change table
cnt=._1 [ lim=. # pairs
while. lim > cnt=.>:cnt do.         NB. process each change pair
  't c'=. cnt { pairs               NB. /target/change
  if. +./b=. t E. y do.             NB. next if no target
    r=. I. b                        NB. target starts
    'l q'=. #&> cnt { pairs         NB. lengths
    p=. r + 0,+/\(<:# r)$ d=. q - l NB. change starts
    s=. * d                         NB. reduce < and > to =
    if. s = _1 do.
      b=. 1 #~ # b
      b=. ((l * # r)$ 1 0 #~ q,l-q) (,r +/ i. l)} b
      y=. b # y
      if. q = 0 do. continue. end.  NB. next for deletions
    elseif. s = 1 do.
      y=. y #~ >: d r} b            NB. first target char replicated
    end.
    y=.(c $~ q *# r) (,p +/i. q)} y  NB. insert replacements
  end.
end. y                              NB. altered string
)

charsub=: 4 : 0

NB.*charsub v-- single character pair replacements.
NB.
NB. dyad:  clPairs charsub cu
NB.
NB.   '-_$ ' charsub '$123 -456 -789'

'f t'=. ((#x)$0 1)<@,&a./.x
t {~ f i. y
)

cnnodesort=: 4 : 0

NB.*cnnodesort v-- class numeric node sort.
NB.
NB. This sort groups euclid digraph  nodes into classes based  on
NB. 'cdp"'     ("c"ommon   notion,   "d"efinition,    "p"ostulate,
NB. "proposition) and  then  sorts  within each group  by numeric
NB. suffix order.
NB.
NB. dyad:  btct =. clClass cnnodesort btctNodes
NB.
NB.   'cdp"' cnnodesort ct

NB. node text and header
t=. ljust&.> s=. }.&.> y [ h=. {.&.> y

NB. check connection prefixes (x)
'invalid connection prefixes' assert *./ ; ({."1 &.> t) e.&.> <x

NB. group nodes into classes
s=. (x&i.&.> {."1 &.> t) </.&.> s

NB. order group classes
s=. s {&.>~ /:@(x&i.)&.> ;&.> ({.@,) L: 0 s

NB. sort incoming nodes only on numeric node suffix
g=. -."1&(a.-.'0123456789')
s=. s ({ L: 0)~ (/:@:".@g) L: 0 ('->'&beforestr"1) L: 0 s

NB. reattach headers
h ,&.> ;&.> s
)


NB. character table to newline delimited list
ctl=: }.@(,@(1&(,"1)@(-.@(*./\."1@(=&' '@])))) # ,@((10{a.)&(,"1)@]))


cutnestidx=: 4 : 0
```

```
NB.*cutnestidx v-- cut list into nested runs and other.
NB.
NB. Nested runs are delimited by begin and end tags. This verb is
NB. oriented toward XML parsing where typical begin  end tags are
NB. <ul>  </ul>  and tags  with  attributes  like: <hoo  boy="2">
NB. </hoo>
NB.
NB. This verb can process numeric lists but care must be taken to
NB. insure the pad item (1{.0$y) does  not  match begin and  end
NB. values.
NB.
NB. dyad:  (ilIdx ;< blcl) =. (clStart;clEnd) cutnestidx cl
NB.        (ilIdx ;< blnl) =. (nlStart;nlEnd) cutnestidx nl
NB.
NB.   xml=. 'yada <ol><li>one</li><ol><li>sub one</li></ol></ol> boo'
NB.   ('<ol';'</ol>') cutnestidx xml
NB.
NB.   88 99 cutnestidx (i.5),88,(10?10),99 88 5 5 5 5 5 99

if. #y do.
  's e'=. ,&.> x               NB. start end lists
  ut=. 1{.0$y                  NB. padding
  assert. -.s -: e             NB. they must differ
  assert. -.(s -:ut) +. e-:ut
  sp=. s E. ut=.y,ut           NB. start mask

  NB. quit if no delimiters
  if. -.1 e. sp do. (i.0);<<y return. end.

  ep=. e E. ut                  NB. end mask
  assert. (+/sp) = +/ep         NB. basic balance
  dp=. sp + - ep                NB. start end marks
  assert. 0 *./ . <: +/\ dp     NB. nested balance
  ep=. I. _1=dp [ sp=. I. 1=dp  NB. start end indexes
  ut=. +/\dp -. 0               NB. scanned marks
  dp=. /:~ sp,ep                NB. all indexes
  sp=. (firstones 1<:ut)#dp     NB. starts of nested
  ep=. (#e)+(0=ut)#dp           NB. starts of other
  dp=. /:~ ~.0,sp,ep            NB. cut starts
  ut=. }: 1 dp} (>:#y)#0        NB. cut mask
  (dp i. sp);<ut <;.1 y         NB. nest indexes cut list
else.
  (i.0);<<y                     NB. empty arg result
end.
)

NB. enclose all character lists in blcl in " quotes
dblquote=: '"'&,@:(,&'"')&.>

eucjoycebkdeps=: 3 : 0

NB.*eucjoycebkdeps v-- all justifications from Joyce book html files.
NB.
NB. monad:  btcl =. eucjoycebkdeps blclHtmlFiles
NB.
NB.   bk1=. 1 dir '~temp/bookI/propI*.html'
NB.   eucjoycebkdeps bk1
NB.
NB.   NB. justifications in first three books
NB.   bks=. 'bookI/propI*html';'bookII/propII*.html';'bookIII/propIII*.html'
NB.   eucjoycebkdeps ; 1&dir&.> (<'~temp/') ,&.> bks

;(<@justfile@winpathsep ,. eucjoycejust@read)&.> y
)

eucjoycecncts=: 3 : 0

NB.*eucjoycecncts v-- format Joyce node connections.
NB.
NB. monad:  blcl =. eucjoycecncts btclPropJust
NB.
NB.   NB. first six books - Byrne's original
NB.   bks=. 'bookI/propI*.html';'bookII/propII*.html';'bookIII/propIII*.html'
NB.   bks=. bks , 'bookIV/propIV*.html';'bookV/propV*.html';'bookVI/propVI*.html'
NB.
NB.   pn=. eucjoycebkdeps ; 1&dir&.> (<'~temp/') ,&.> bks
NB.   cn=. eucjoycecncts pn
NB.
NB.   NB. assemble graphviz dot
NB.   hdr=. 4 disp 'joyce_graphviz_header_gv'
NB.   gv=. hdr,(2#LF),(ctl > ~.cn),(2#LF),'}'
NB.   (toHOST gv) write jpath '~temp/euclid_joyce_1_3.gv'
NB. clean up some "tickyboos"
pj=. '/Cor./Cor/. /./ /.'&changestr@(((', '&charsub)@rebc)@allwhitetrim)&.> y

NB. standardize proposition names
pn=. '.'&beforelaststr&.> (#'prop')&}.&.> 0 {"1 pj
ix=. pn i.&1@e.&> <'0123456789'
pn=. (ix {.&.> pn) ,&.> '.' ,&.> ix }.&.> pn

NB. remove any trailing periods
pj=. pj }.&.>~ -'.'={:&> pj=. 1 {"1 pj

NB. rename postulates
pstnew=. 2 }.&.> pstold=. <;._1 ' I.Post.1 I.Post.2 I.Post.3 I.Post.4 I.Post.5'
ix=. 0 -.~ <"1 I."1 pj =/~ pstold
for_pr. ix do. pj=. (pr_index{pstnew) (;pr)} pj end.

NB. format graphviz connections
(dblquote pj) ,&.> (<' -> ') ,&.> (dblquote pn) ,&.> ';'
)

eucjoycedeps=: 3 : 0

NB.*eucjoycedeps v-- extract  noted book  dependencies from Joyce
NB. html.
NB.
NB. NOTE: this verb is a dead end. It turns out that Joyce's book
NB. cross  references  only refer to propositions in the  current
NB. book  and  not  across  books.   You  have  to  extract   the
NB. "justifications" from all  the proposition files to go across
NB. books. See ().
NB.
NB. monad:  bt =. eucjoycedeps clHtmlTab
NB.
NB.   NB. fetch text from (futs)
NB.   html=. 4 disp 'Joyce_Elements_Books_I_VI_Html_txt'
NB.   btabs=.  eucjoycetabs html
NB.   ({."1 btabs) ,: >&.> eucjoycedeps L: 0 {:"1 btabs

NB. cut rows and cols
s=. (] <;.1~ '<td>' E. ])&> ('<tr>' E. y) <;.1 y

NB. extract element text
s=. ('a'&geteleattrtext)@rebc&.> s -.&.> <TAB,CR,LF

NB. remove empty rows
s #~ 0 < +/"1 #&> s
)

eucjoycehtml=: 3 : 0

NB.*eucjoycehtml  v-- html from David Joyce's online Elements.
NB.
NB. monad:  clHtml =. eucjoycehtml uuIgnore
NB.
NB.    NB. save web pages as text
NB.    file=. 'Joyce_Elements_Books_I_VI_Html.txt'
NB.    (eucjoycehtml 0) write jpath '~temp/',file
NB.    puttxt file

NB. require 'web/gethttp' !(*)=. gethttp

NB. first six books - only books in Byrne's edition
bk1=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookI/bookI.html'
bk2=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookII/bookII.html'
bk3=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookIII/bookIII.html'
bk4=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookIV/bookIV.html'
bk5=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookV/bookV.html'
bk6=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookVI/bookVI.html'
;bk1;bk2;bk3;bk4;bk5;bk6
)

eucjoycejust=: 3 : 0

NB.*eucjoycejust v-- extract justifications from Joyce proposition html.
NB.
NB. monad:  blcl =. eucjoycejust clHtml
NB.
NB.   htm0=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookI/propI1.html'
NB.   htm1=. gethttp 'https://mathcs.clarku.edu/~djoyce/elements/bookIII/propIII16.html'
NB.   eucjoycejust htm0
NB.   eucjoycejust htm1

NB. justifications in text
```

```
(,'a'&geteleattrtext"1 (] '</div>'&beforestr;.1~ '<div class="just">' E. ]) y -. CR,LF,TAB) -. a:
)

eucjoycetabs=: 3 : 0

NB.*eucjoycetabs v-- extract dependency tables from Joyce html.
NB.
NB. Not all these web  pages have  dependency tables. Extract the
NB. extant tables.
NB.
NB. monad:  bt =. eucjoycetabs clHtml
NB.
NB.   NB. fetch text from (futs)
NB.   html=. 4 disp 'Joyce_Elements_Books_I_VI_Html_txt'
NB.   btabs=. eucjoycetabs html

nada=. 0 2$a: NB. no tables

NB. cut into pages
bks=. (] <;.1~ '<HTML><HEAD>' E. ]) y

NB. pages with tables
if. -. +./tbs=. +./@('</table>'&E.)&> bks do. nada
else.

  NB. all tables on pages
  bks=. {{ '</table>'&beforestr&.> ('<table ' E. y) <;.1 y }} &.> tbs#bks
  tbs=. >: I. tbs  NB. elements book numbers

  NB. only page dependency tables
  q=. +./&> p=. ;&.> +./@('Dependencies within'&E.) L: 0 bks
  if. -. +./q do. nada return. end.

  NB. book numbers with tables
  tbs=. q # tbs [ bks=. q # p #&.> bks
  (<"0 tbs) ,. bks
end.
)

eucnctsparse=: 3 : 0

NB.*eucnctsparse v-- parses euclid digraph gv code.
NB.
NB. Splits digraph  code into  preamble,  postamble and a  unique
NB. table of sorted connections.
NB.
NB. monad:  bl =. eucnctsparse clGv
NB.
NB.   NB. dot digraph code in (futs)
NB.   gv=. read dotgv_ijod_=. getbyte 'euclid_joyce_1_6_b_gv'
NB.   eucnctsparse gv

bI=. eucSTARTBOOKDEPS [ eI=. eucENDBOOKDEPS
nbI=. eucSTARTNODEATTRS [ neI=. eucENDNODEATTRS
'node connection delimiters' assert (1 = +/bI E. y) *. 1 = +/eI E. y
```

```
'node attribute delimiters' assert (1 = +/nbI E. y) *. 1 = +/neI E. y

NB. preamble and postamble
pr=. bI beforestr y [ po=. allwhitetrim eI,eI afterstr y

NB. remove old node attributes
pr=. allwhitetrim nbI beforestr pr

NB. book nodes
c=. CR -.~ tlf eI beforestr bI afterstr y
c=. (<'"; ') -.&.>~ ('->'&beforestr ; '->'&afterstr);._1 tlf c -.CR
c=. c #~ *./"1 ] 0 < #&> c

NB. sort by Euclid book and numeric proposition
NB. number and make connections unique
s=. >('.'&beforestr ; '.'&afterstr )&.> 1 {"1 c
c=. ~. c {~ /: (RomanElementsBooks i. 0 {"1 s) ,. ".&> 1 {"1 s
'node self loop(s)' assert 0 = +/ =/"1 c

NB. preamble, postamble, connections
pr;po;<c
)

eucsortBgv=: 3 : 0

NB.*eucsortBgv v-- second sort and format euclid book digraphs.
NB.
NB. WARNING: this verb expects a particular graph text layout.
NB.
NB. monad:  cl =. eucsortBgv clGv
NB.
NB.   NB. dot digraph code in (futs)
NB.   gv=. read dotgv_ijod_=. getbyte 'euclid_joyce_1_6_b_gv'
NB.
NB.   NB. typical use
NB.   ngv=. eucsortBgv gv
NB.   (toHOST ngv) write dotgv_ijod_
NB.   graphview dotgv_ijod_

bI=. eucSTARTBOOKDEPS [ eI=. eucENDBOOKDEPS
nbI=. eucSTARTNODEATTRS [ neI=. eucENDNODEATTRS

'pr po c'=. eucnctsparse y

NB. main site url
urh=. 'https://mathcs.clarku.edu/~djoyce/elements/book'

NB. terminal nodes - end of the trail cowboy
t=. (~.,c) -. 0 {"1 c
t=. (dblquote t) ,&.> <' [fillcolor=',ncolorTERMINAL,'];'
tends=. LF,('// terminal nodes',LF) , ;t ,&.> LF

NB. postulate node attributes
p=. c #~ +./@('Post.'&E.)&> 0 {"1 c
p=. /:~ p #~ ~: 0 {"1 p
```

```
purl=. tolower&.> ((0 {"1 p) -.&.> '.') ,&.> <'.html"];'
purl=. (<' [fillcolor=',ncolorPOSTULATE,', URL="',urh,'I/') ,&.> purl
post=. (dblquote 0 {"1 p) ,&.> purl
post=. LF,('// postulates',LF),ctl ;post ,&.> LF

NB. common notions
cnurl=. ' [fillcolor=',ncolorNOTION,', URL="',urh,'I/cn.html"];'
cn=. c #~ +./@('C.N'&E.)&> 0 {"1 c
cn=. /:~ cn #~ ~: 0 {"1 cn
comn=. (dblquote 0 {"1 cn) ,&.> <cnurl
comn=. LF,('// common notions',LF),ctl ;comn ,&.> LF

NB. definition node attributes
d=. c #~ +./@('.Def.'&E.)&> 0 {"1 c
d=. d #~ ~: 0 {"1 d
NB. NOTE: the links to definitions are not one-to-one
def=. (dblquote 0 {"1 d) ,&.> <' [fillcolor=',ncolorDEFINITION,'];'
def=. LF,('// definitions',LF),ctl ;def ,&.> LF

NB. proposition node attributes
prop=. ~. 1 {"1 c
t=. (<'/prop') ,&.> (prop -.&.> '.') ,&.> <'.html"];'
pruh=. <' [URL="',urh
prop=. (dblquote prop) ,&.> pruh ,&.> ('.'&beforestr&.> prop) ,&.> t
prop=. LF,('// propositions',LF),ctl ;prop ,&.> LF

NB. reassemble
```

```
natt=. nbI,(2#LF),(allwhitetrim tends,post,comn,def,prop),(2#LF),neI
c=. 0 2 1 {"1 (dblquote c) ,"1 <' -> '
c=. (0 1 {"1 c) ,. (2 {"1 c) ,&.> ';'
pr,(2#LF),natt,(2#LF),bI,LF,(ctl ;"1 c),LF,po
)

eucsortgv=: 3 : 0

NB.*eucsortgv v-- sort and format euclid book digraphs.
NB.
NB. Sort of  incoming Euclid  Book  graphviz  digraph nodes.  The
NB. order is  ignored  by  graphviz  but  it makes it  easier  to
NB. inspect the graphs.
NB.
NB. WARNING: this verb expects a particular graph text layout.
NB.
NB. monad:  cl =. eucsortgv clGv
NB.
NB.   NB. digraph DOT text in (futs)
NB.   NB. places (euclid_1.gv) in J temp
NB.   getbyte 'euclid_1_gv'
NB.
NB.   NB. typical use
NB.   gv=. jpath '~temp/euclid_1.gv'
NB.   (toHOST st=. eucsortgv read gv) write gv
NB.   graphview gv
NB.   putbyte 'euclid_1.gv'
bI=. eucSTARTBOOKDEPS [ eI=. eucENDBOOKDEPS
'node delimiters' assert (1 = +/bI E. y) *. 1 = +/eI E. y

NB. preamble and postamble
pr=. bI beforestr y [ po=. eI,eI afterstr y

NB. book nodes
c=. CR -.~ tlf eI beforestr bI afterstr y

NB. cut nodes
c=. (1 (0)} '//---' E. c) <;.1 c

NB. table all but first item
ct=. rebrow&.> (];._2)&.> }. c

NB. alpha sort node tables
NB. ct=. ctl ; ' ' ,&.> (0 ,&.> >:@/:&.> (tolower@}.&.> ct) -."1&.> <'" ') {&.> ct

NB. numeric prefix grouped sort
ct=. ctl ; ' ' ,&.> 'cdp"' cnnodesort ct

NB. reassemble
(allwhitetrim pr,bI),(2#LF),(allwhitetrim ct),(2#LF),allwhitetrim po
)

NB. 0's all but first 1 in runs of 1's - like (firstone) but differs for nulls
firstones=: > (0: , }:)
```

```
NB. get element text following attributes
geteleattrtext=: [: '>'&afterstr&.> ] betweenstrs~ [: atags [: alltrim [

gvclustoff=: 3 : 0

NB.*gvclustoff v-- dot code marked cluster(s) off.
NB.
NB. monad:  gvclustoff ??
NB. dyad:  ?? gvclustoff ??

NB. check for off marks
'bCl eCl'=. CLUSTOFFMARKS
'dot clusters off' assert (0=+/bCl E. y) *. 0=+/eCl E. y

NB. on marks
'bCl eCl'=. 2 }.&.> CLUSTOFFMARKS
'on dot clusters bad' assert (0 < c) *. (+/bCl E. y) = c=. +/eCl E. y

NB. cut out on clusters
'ix ct'=. (bCl;eCl) cutnestidx y

NB. turn them off and reassemble
; ({{ ctl '//' ,"1 ];._2 tlf y -. CR }} &.> ix{ct) ix} ct
)

gvcluston=: 3 : 0
```

```
NB.*gvcluston v-- dot code marked cluster(s) on.
NB.
NB. monad:  cl =. gvcluston clDot
NB.
NB.   gv=. read getbyte 'euclid_1_2_gv'
NB.   dotgv_ijod_=: jpath '~temp/test.gv'
NB.   (toHOST gvcluston gv) write dotgv
NB.   graphview dotgv
NB.
NB.   NB. throws assertion
NB.   gvcluston gvcluston gv

'bCl eCl'=. CLUSTOFFMARKS
'off dot clusters bad' assert (0 < c) *. (+/bCl E. y) = c=. +/eCl E. y

NB. cut out off clusters
'ix ct'=. (bCl;eCl) cutnestidx y

NB. turn them on and reassemble
; ({{ ctl '//'&afterstr"1 ];._2 tlf y -. CR }} &.> ix{ct) ix} ct
)


NB. file name from fully qualified file names
justfile=: (] #~ [: -. [: +./\ '.'&=)@(] #~ [: -. [: +./\. e.&':\')

NB. left justify table
ljust=: ' '&$: :(] |."_1~ i."1&0@(] e. [))
```

```
NB. reads a file as a list of bytes
read=: 1!:1&(]`<@.(32&>@(3!:0)))

NB. removes multiple blanks (char only)
rebc=: ] #~ [: -. '  '&E.

NB. deletes all blank rows from character table
rebrow=: ] #~ [: -. [: *./"1 ' '&=

NB. appends trailing line feed character if necessary
tlf=: ] , ((10{a.)"_ = {:) }. (10{a.)"_

tolower=: 3 : 0

NB.*tolower v-- convert to lower case.
NB.
NB. monad: cl =. tolower cl

x=. I. 26 > n=. ((65+i.26){a.) i. t=. ,y
($y) $ ((x{n) { (97+i.26){a.) x}t
)

NB. standardizes path delimiter to windows back \ slash
winpathsep=: '\'&(('/' I.@:= ])} )

NB.POST_eucgvuts post processor.
smoutput IFACE=:  (0 : 0)
NB. (eucgvuts) interface word(s): 20230623j132822
NB. --------------------------
NB. eucjoycebkdeps  NB. all justifications from Joyce book html files
NB. eucjoycecncts   NB. format Joyce node connections
NB. eucjoycedeps    NB. extract noted book dependencies from Joyce html
NB. eucjoycehtml    NB. html from David Joyce's online Elements
NB. eucjoycetabs    NB. extract dependency tables from Joyce html
NB. eucsortBgv      NB. second sort and format euclid book digraphs
NB. eucsortgv       NB. sort euclid book digraphs
NB. gvclustoff      NB. dot code marked cluster(s) off
NB. gvcluston       NB. dot code marked cluster(s) on
)

cocurrent 'base'
coinsert  'eucgvuts'
```

Index