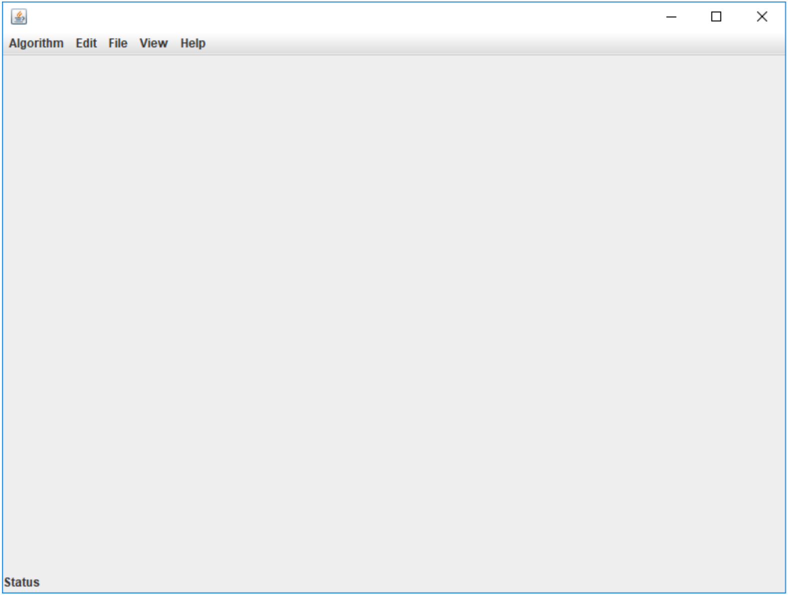
fancy software name

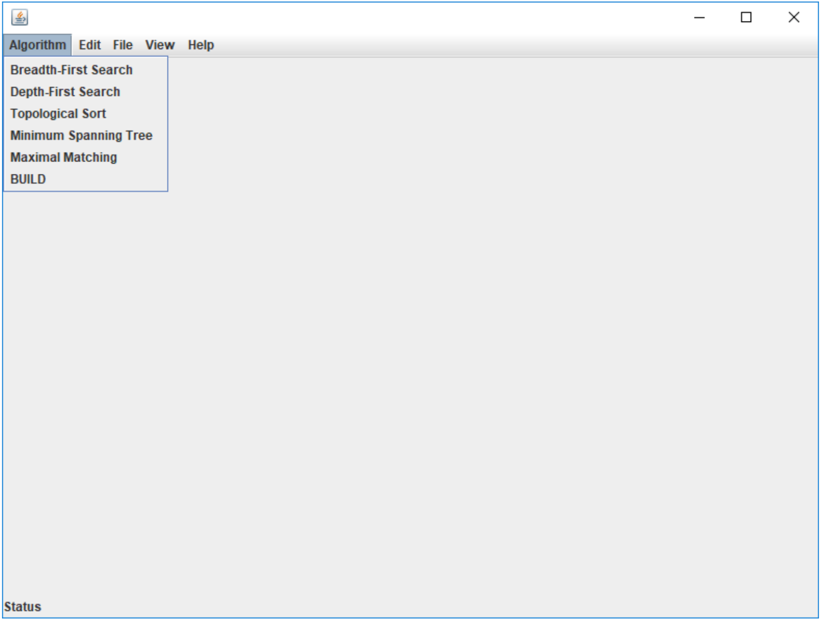
Anica Hoppe, Sonja Türpitz

May 2, 2018

GUI



Algorithms


Algorithm
Edit
File
View
Help
Breadth-First Search
Depth-First Search
Topological Sort
Minimum Spanning Tree
Maximal Matching
BUILD
Status

Breadth/Depth-First Search

Input: $G = (V, E),\ s \in V$

$Q, F \leftarrow \emptyset$

$R \leftarrow \{s\}$ $\triangleright$ visited vertices

for all $(s, v) \in E$ **do**

  $Q \leftarrow Q + (s, v)$

while $Q \neq \emptyset$ **do**

  $(v, w) \leftarrow \mathit{choose}(Q)$ $\triangleright$ Q is queue or stack (BFS/DFS)

  if $w \notin R$ or $v \notin R$ **then**

    $R \leftarrow R + w + v$

    $F \leftarrow F + (v, w)$

    for all $(w, x) \in E$ **do**

      $Q \leftarrow Q + (w, x)$

return (V, F)

source: lecture graphtheory by Prof. Dr. Marc Hellmuth

Topological Sort

```
L ← empty list                              ▷ will contain the sorted elements
S ← set of all vertices with no incoming edge
while S ≠ ∅ do
    remove a vertex n from S
    add n to tail of L
    for all vertices m with an edge e from n to m do
        remove edge e from the graph
        if m has no other incoming edge then
            insert m into S
if graph has edges then
    return error                            ▷ graph has at least one cycle
else
    return L
```

source: https://en.wikipedia.org/wiki/Topological_sorting

Minimum Spanning Tree

$F \leftarrow \emptyset$
for $i = 1$ to $|E|$ **do**
    if $(V, F + e_i)$ is acyclic **then**
        $F \leftarrow F + e_i$
return (V, F)

source: lecture graphtheory by Prof. Dr. Marc Hellmuth

Maximal Matching

$M \leftarrow \emptyset$
$E' \leftarrow E$ ▷ not visited edges
while $E' \neq \emptyset$ **do**
  choose $e = (u, v) \in E'$
  $M \leftarrow M + (u, v)$
  delete all edges from E' that are incident to u or v
return M

source: lecture discret optimization by Prof. Dr. Marc Hellmuth, own

BUILD

BUILD

INPUT: Set of triples in $\mathscr{R}$, leaf set $\mathscr{L}$.
OUTPUT: A rooted, phylog. tree distinctly leaf-labeled by $\mathscr{L}$ consistent with all rooted triplets in $\mathscr{R}$, if one exists; otherwise *null*.
compute $G(\mathscr{R},\mathscr{L})$
compute connected components $C_1,\dots,C_s$ of $G(\mathscr{R},\mathscr{L})$
if $s=1$ and $|\mathscr{L}|=1$ **then**
    return tree $\simeq K_1$
else if $s=1$ and $|\mathscr{L}|>1$ **then**
    return *null*
else
    for $i=1,\dots s$ **do**
        $T_i = \mathtt{BUILD}(\mathscr{R}_{|V(C_i)}, V(C_i))$
    end for
    if $T_i \neq null$ for all $i=1,\dots s$ **then**
        attach all of these trees to a common parent node and let T be the resulting tree; else $T = null$.
    end if
end if

source: lecture bioinformatics by Prof. Dr. Marc Hellmuth

Requirements

properties of the graph

- no coloring of edges or vertices
- no weights of edges or vertices

  $\rightarrow$ not needed for algorithms
- mainly undirected graphs
- vertex label

  $\rightarrow$ for BUILD

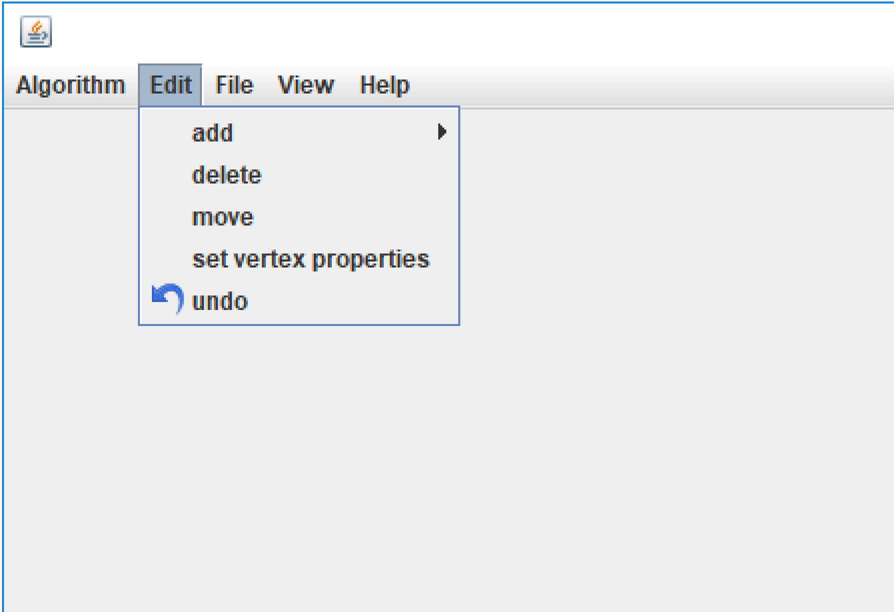
$\rightsquigarrow$ as easy as possible

Requirements

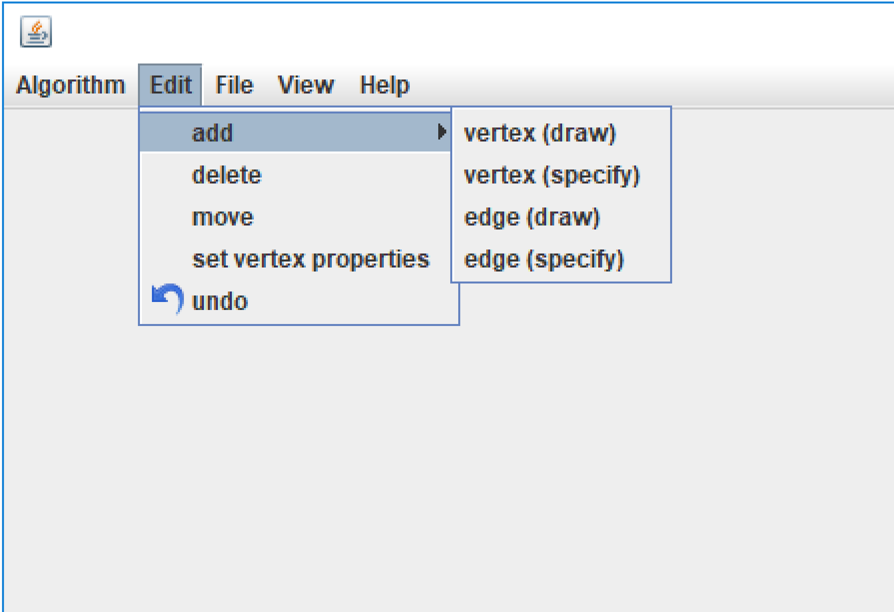
format of textfile

1	'u' or 'd' (undirected or directed)
2	Integer n (number of vertices)
3	Integer m (number of edges)
$4 - (n+4)$	vertex list (name, label) for each vertex
$(n+5) - (m+n+5)$	edge list (u,v) for each edge

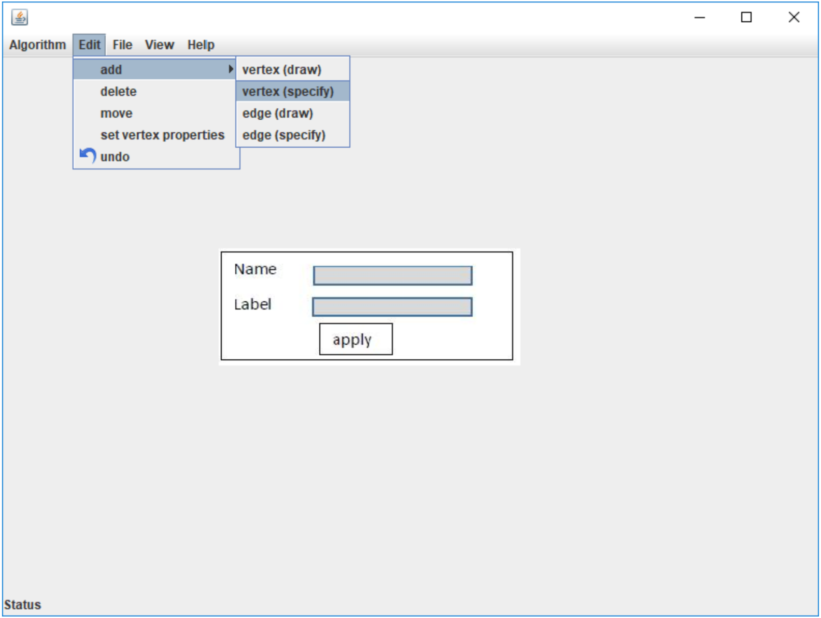
Menues


Algorithm
Edit
File
View
Help
add
delete
move
set vertex properties
undo

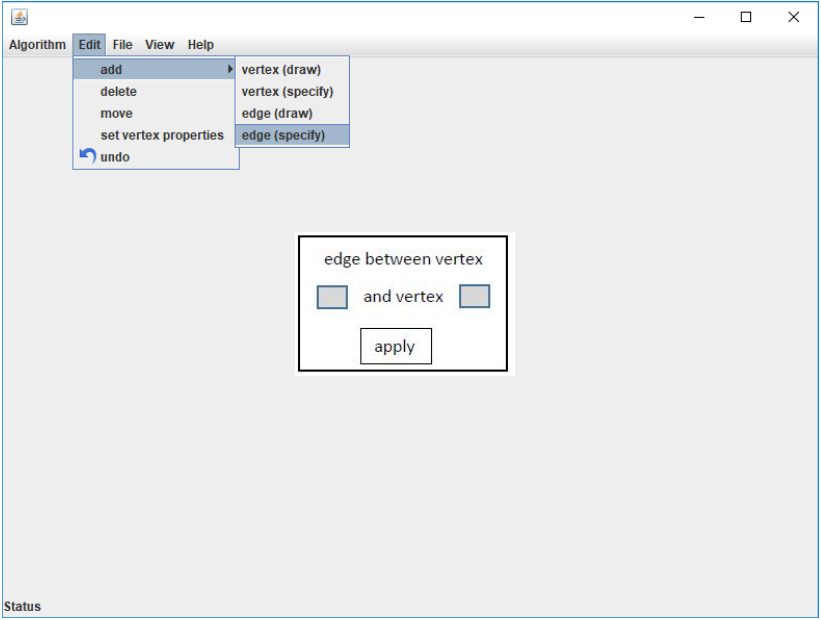
Menues


Algorithm
Edit
File
View
Help
add
delete
move
set vertex properties
undo
vertex (draw)
vertex (specify)
edge (draw)
edge (specify)

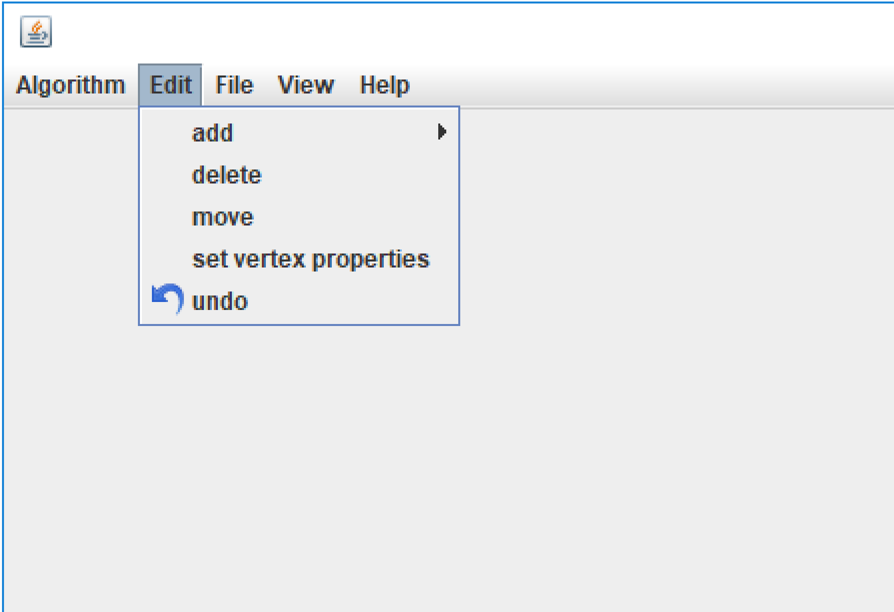
Menues


Algorithm
Edit
File
View
Help
add
delete
move
set vertex properties
undo
vertex (draw)
vertex (specify)
edge (draw)
edge (specify)
Name
Label
apply
Status

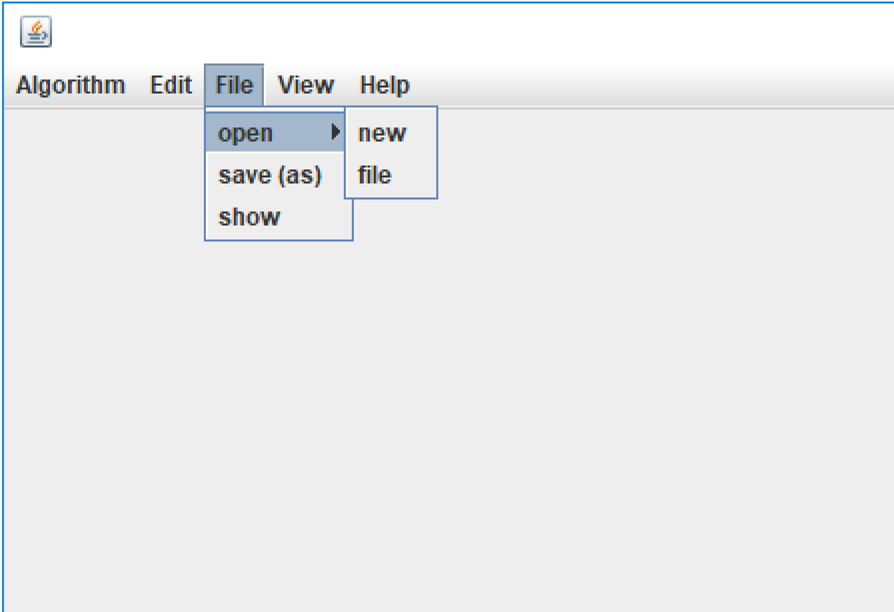
Menues


Algorithm
Edit
File
View
Help
add
delete
move
set vertex properties
undo
vertex (draw)
vertex (specify)
edge (draw)
edge (specify)
edge between vertex
and vertex
apply
Status

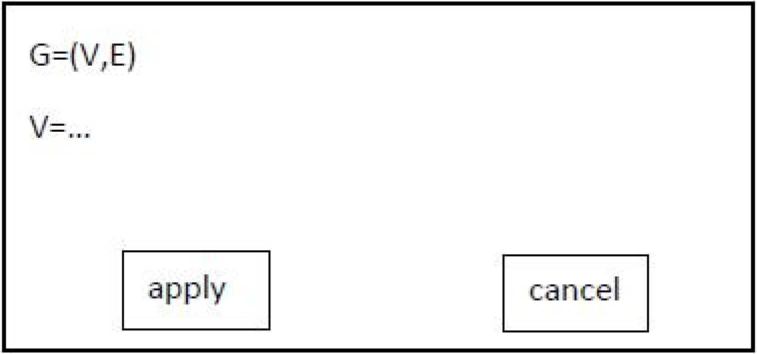
Menues


Algorithm
Edit
File
View
Help
add
delete
move
set vertex properties
undo

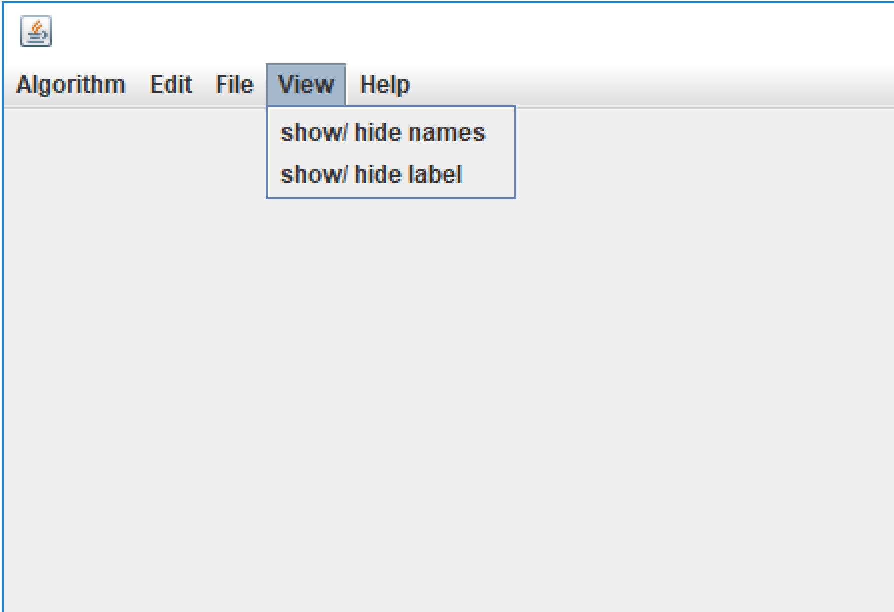
Menues



Menues



Menues


Algorithm
Edit
File
View
Help
show/ hide names
show/ hide label

Thank you for your attention!