There are the following JAVA classes implementing the multi-frontal direct solver algorithm for finite difference problem:

- *A.java*
- *A1.java*
- *A2.java*
- *AN.java*
- *Aroot.java*
- *BS.java*
- *E2.java*
- *Eroot.java*
- *Executor.java*
- *Main.java*
- *P1.java*
- *P2.java*
- *P3.java*
- *Production.java*
- *Vertex.java*

We have already implemented all the classes for the exemplary solver execution for finite difference method with workflow corresponding to the mesh with 6 elements, as it is presented in Figures 1 and 2.

Now, we are going to replace the code to work with time-dependent problem, in particular for the α-scheme with respect to time mixed with finite element method with linear basis functions for space

Please replace system in *A1* by

$$\begin{bmatrix} \frac{h}{3} + \Delta t\ \alpha \left(\frac{1}{h} + 1\right) & \frac{h}{6} - \Delta t\ \alpha\ \frac{1}{h} \\ \frac{h}{6} - \Delta t\ \alpha\ \frac{1}{h} & \frac{h}{3} + \Delta t\ \alpha\ \frac{1}{h} \end{bmatrix} \begin{bmatrix} a_0^{t+1} \\ a_1^{t+1} \end{bmatrix} = \begin{bmatrix} \left\{\frac{h}{3} - \Delta t\ \beta \left(\frac{1}{h} + 1\right)\right\} a_0^t + \left(\frac{h}{6} + \Delta t\ \beta\ \frac{1}{h}\right) a_1^t + \Delta t \\ \left(\frac{h}{6} + \Delta t\ \beta\ \frac{1}{h}\right) a_0^t + \left(\frac{h}{3} - \Delta t\ \beta\ \frac{1}{h}\right) a_1^t \end{bmatrix}$$

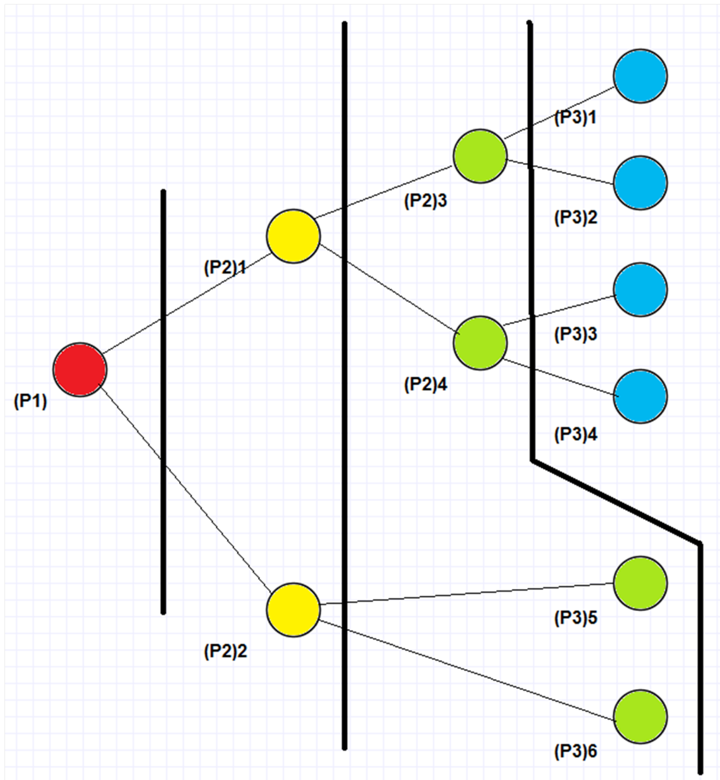

Figure 1: Graph of tasks responsible for construction of the elimination tree over the mesh with 6 elements.

replace system in *A* by

$$\begin{bmatrix} \frac{h}{3} + \Delta t \; \alpha \, \frac{1}{h} & \frac{h}{6} - \Delta t \; \alpha \, \frac{1}{h} \\ \frac{h}{6} - \Delta t \; \alpha \, \frac{1}{h} & \frac{h}{3} + \Delta t \; \alpha \, \frac{1}{h} \end{bmatrix} \begin{bmatrix} a_1^{t+1} \\ a_2^{t+1} \end{bmatrix} = \begin{bmatrix} \left(\frac{h}{3} - \Delta t \; \beta \frac{1}{h}\right) a_1^t + \left(\frac{h}{6} + \Delta t \; \beta \, \frac{1}{h}\right) a_2^t \\ \left(\frac{h}{6} + \Delta t \; \beta \, \frac{1}{h}\right) a_1^t + \left(\frac{h}{3} - \Delta t \; \beta \, \frac{1}{h}\right) a_2^t \end{bmatrix}$$

replace system in *AN* by

$$\begin{bmatrix} \frac{h}{3} + \Delta t \; \alpha \, \frac{1}{h} & \frac{h}{6} - \Delta t \; \alpha \, \frac{1}{h} \\ \frac{h}{6} - \Delta t \; \alpha \, \frac{1}{h} & \frac{h}{3} + \Delta t \; \alpha \left(\frac{1}{h} + 1\right) \end{bmatrix} \begin{bmatrix} a_{n-1}^{t+1} \\ a_n^{t+1} \end{bmatrix} = \begin{bmatrix} \left(\frac{h}{3} - \Delta t \; \beta \, \frac{1}{h}\right) a_{n-1}^t + \left(\frac{h}{6} + \Delta t \; \beta \, \frac{1}{h}\right) a_n^t \\ \left(\frac{h}{6} + \Delta t \; \beta \, \frac{1}{h}\right) a_{n-1}^t + \left\{\frac{h}{3} - \Delta t \; \beta \left(\frac{1}{h} + 1\right)\right\} a_n^t - \Delta t \end{bmatrix}$$

To do that, it is necessary to add the element diameter h, the time step dt and α paramter into the Java code. Additionally, the previous time step solution `x_old` must be stored. This should be done in the *Vertex* class:

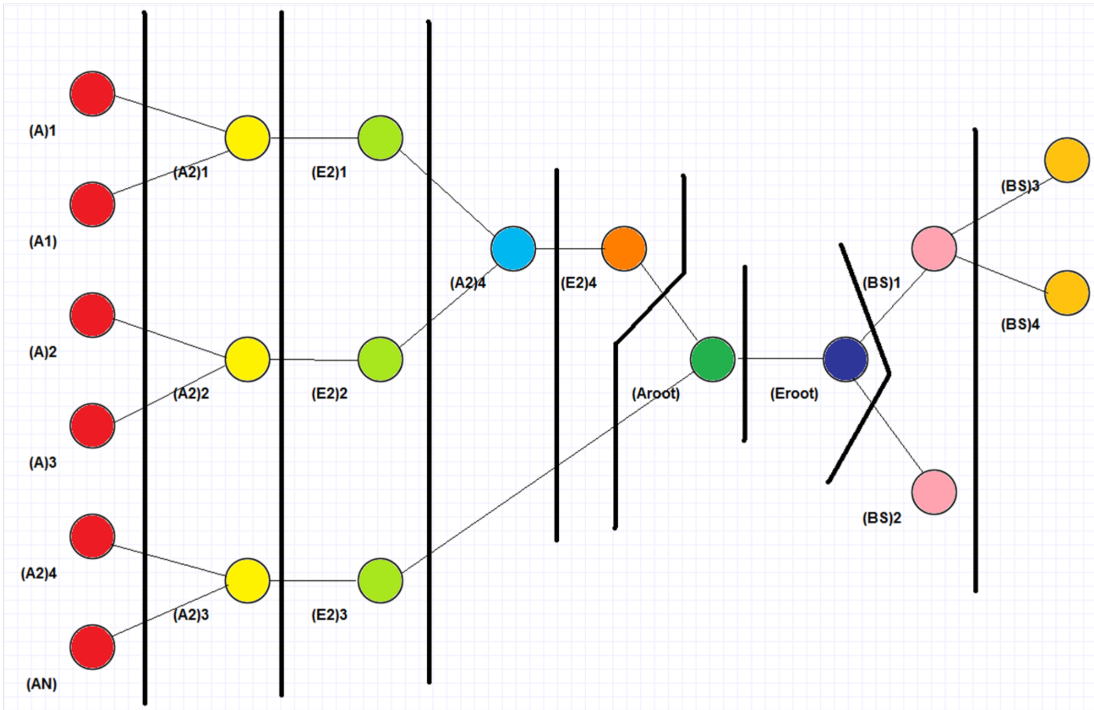

Figure 2: Graph of tasks responsible for execution of the solver algorithm over the constructed elimination tree.

```
class Vertex {
  Vertex(Vertex Left, Vertex Right, Vertex Parent, String Lab){
    this.left=Left;
    this.right=Right;
    this.parent=Parent;
    this.label=Lab;
    a = new double[3][3];
    b = new double[3];
  }
String label;
Vertex left;
Vertex right;
Vertex parent;
double[][] a;
double[] b;
double[] x;
```

```
double[] x_old;
static double h;
static double dt;
static double alpha;
void set_left(Vertex Left){left=Left;}
void set_right(Vertex Right){right=Right;}
void set_parent(Vertex Parent){parent=Parent;}
void set_label(String Lab){label=Lab;}
}
```

Hint: The *A1* updated production looks like this:

```
class A1 extends Production {
  A1(Vertex Vert,CyclicBarrier Barrier){
    super(Vert,Barrier);
  }
Vertex apply(Vertex T) {
  System.out.println(''A'');
  T.a[1][1]=h/3 + dt α (1/h+1);
  T.a[2][1]=h/6 - dt α 1/h;
  T.a[1][2]=h/6 - dt α 1/h;
  T.a[2][2]=h/3 + dt α 1/h;
  T.b[1]=xt[1]*(h/3-dt β (1/h+1))+xt[2]*(h/6+dt β 1/h)+dt;
  T.b[2]=xt[1]*(h/6+dt β 1/h)+xt[2]*(h/3-dt β 1/h);
  return T;
  }
}
```