Алгоритм Левенштейна Trace

```
function lev($source, $dest)
{
        if ($source == $dest) {
    return 0;
  }

  list($slen, $dlen) = [strlen($source), strlen($dest)];

  if ($slen == 0 || $dlen == 0) {
    return $dlen ? $dlen : $slen;
  }
        $dist[0][0]=0;
        for($i=1;$i<=$slen;$i++)
        {
                $dist[$i][0]=$dist[$i-1][0]+1;

                //$from[$i][0][0]=$i;
                //$from[$i][0][1]=0;
                //$from[$i][0][2]=1;
        }
        for($j=1;$j<=$dlen;$j++)
        {
                $dist[0][$j]=$dist[0][$j-1]+1;

                //$from[0][$j][0]=0;
                //$from[0][$j][1]=$j;
                //$from[0][$j][2]=1;
        }
        for($i=1;$i<=$slen;$i++)
        {
                for($j=1;$j<=$dlen;$j++)
                {

                        $cost = ($source[$i-1] == $dest[$j-1]) ? 0 :
1;
                        //$dist[$i][$j]=mymin(
                        //      $dist[$i][$j-1]+1,
                        //      $dist[$i-1][$j]+1,
                        //      $dist[$i-1][$j-1]+$cost
                        //);
                        $result=0;
                        $dist[$i][$j]=mymin(
                                $dist[$i][$j-1]+1,
                                $dist[$i-1][$j]+1,
                                $dist[$i-1][$j-1]+$cost,
                                $result
                        );
                        //if($dist[$i][$j]==$dist[$i][$j-1]+1)
                        if($result==1)
                        {
                                $from[$i][$j][0]=$i;
                                $from[$i][$j][1]=$j-1;
                                $from[$i][$j][2]=1;
                        }
                        //if($dist[$i][$j]==$dist[$i-1][$j]+1)
                        if($result==2)
                        {
                                $from[$i][$j][0]=$i-1;
                                $from[$i][$j][1]=$j;
                                $from[$i][$j][2]=2;
                        }
                        //if($dist[$i][$j]==$dist[$i-1][$j-1]+$cost)
                        if($result==3)
                        {
                                $from[$i][$j][0]=$i-1;
                                $from[$i][$j][1]=$j-1;
                                $from[$i][$j][2]=3;
                        }
                }
        }
        //echo '<pre>';
        //print_r($from);
        //echo '</pre>';
        //echo '-----------------------------------------------------<br>';
        //echo '<pre>';
        //print_r($dist);
        //echo '</pre>';
        //printarr($dist,$slen,$dlen);
        echo '<br>';
        //print_from($from, $slen-1, $dlen-1, $source, $dest);
        print_lgc($dist, $source, $dest);
        return $dist[$slen][$dlen];
}
```

```
//Longest common subsequence problem
function print_lgc($dist,$source,$dest)
{
        list($slen, $dlen) = [strlen($source), strlen($dest)];
        $i = $slen;
        $j = $dlen;

        $nakopitel=[];
        $n=0;

        while($i > 0 || $j > 0)
        {
          if($dist[$i][$j] == $dist[$i-1][$j] + 1)
          {
                $i = $i - 1;
                $del[] = $i;
          }
          else if($dist[$i][$j] == $dist[$i][$j-1] + 1)
          {
                $j = $j - 1;
                $ins[] = $j;
          }
          else
          {
                //echo '('.$i.','.$j.')';
                $nakopitel[$n][]=$i;
                $nakopitel[$n][]=$j;
                $n++;

                $sub['i'][] = $i;
    $sub['j'][] = $j;
                $i = $i - 1;
                $j = $j - 1;
          }
          //$i = $i - 1;
          //$j = $j - 1;
          //echo '('.$i.','.$j.')';
        }
        echo '<br>';

        echo '---------------------------------------<br>';
        echo 'del:<br>';
        echo '<pre>';
        print_r($del);
        echo '</pre>';

        echo '---------------------------------------<br>';
        echo 'ins:<br>';
        echo '<pre>';
        print_r($ins);
        echo '</pre>';

        echo '---------------------------------------<br>';
        echo 'sub:<br>';
        echo '<pre>';
        print_r($sub);
        echo '</pre>';

        echo '---------------------------------------<br>';
        echo 'nakopitel:<br>';
        echo '<pre>';
        print_r($nakopitel);
        echo '</pre>';

        for($k=count($nakopitel)-2;$k>=0;$k--)
        {
                echo $source[$nakopitel[$k][0]-1].' ';
        }
        echo '<br>';
        for($k=count($nakopitel)-2;$k>=0;$k--)
        {
                echo $dest[$nakopitel[$k][1]-1].' ';
        }
        echo '<br>';
        //$source,$dest
}
```