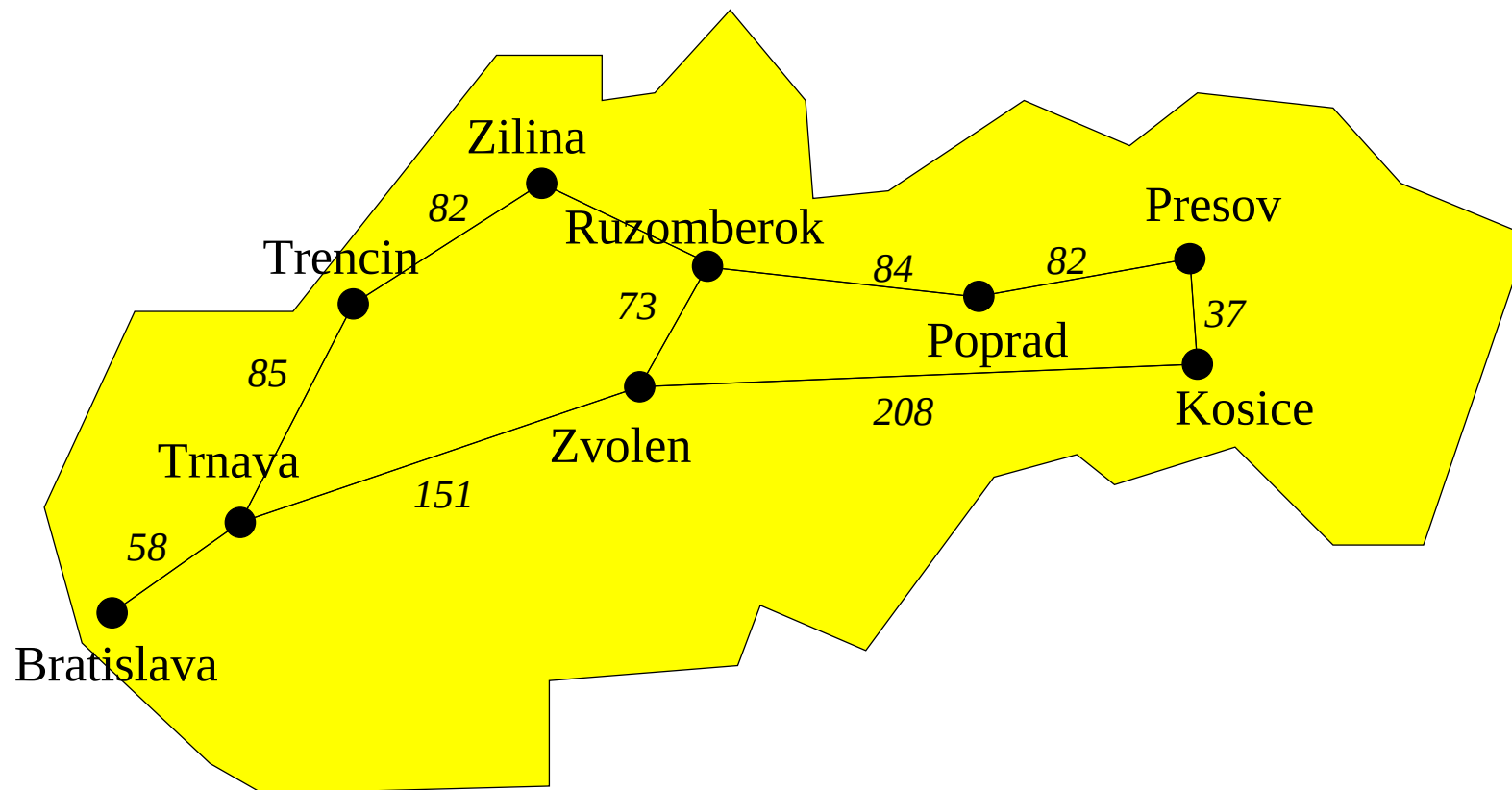
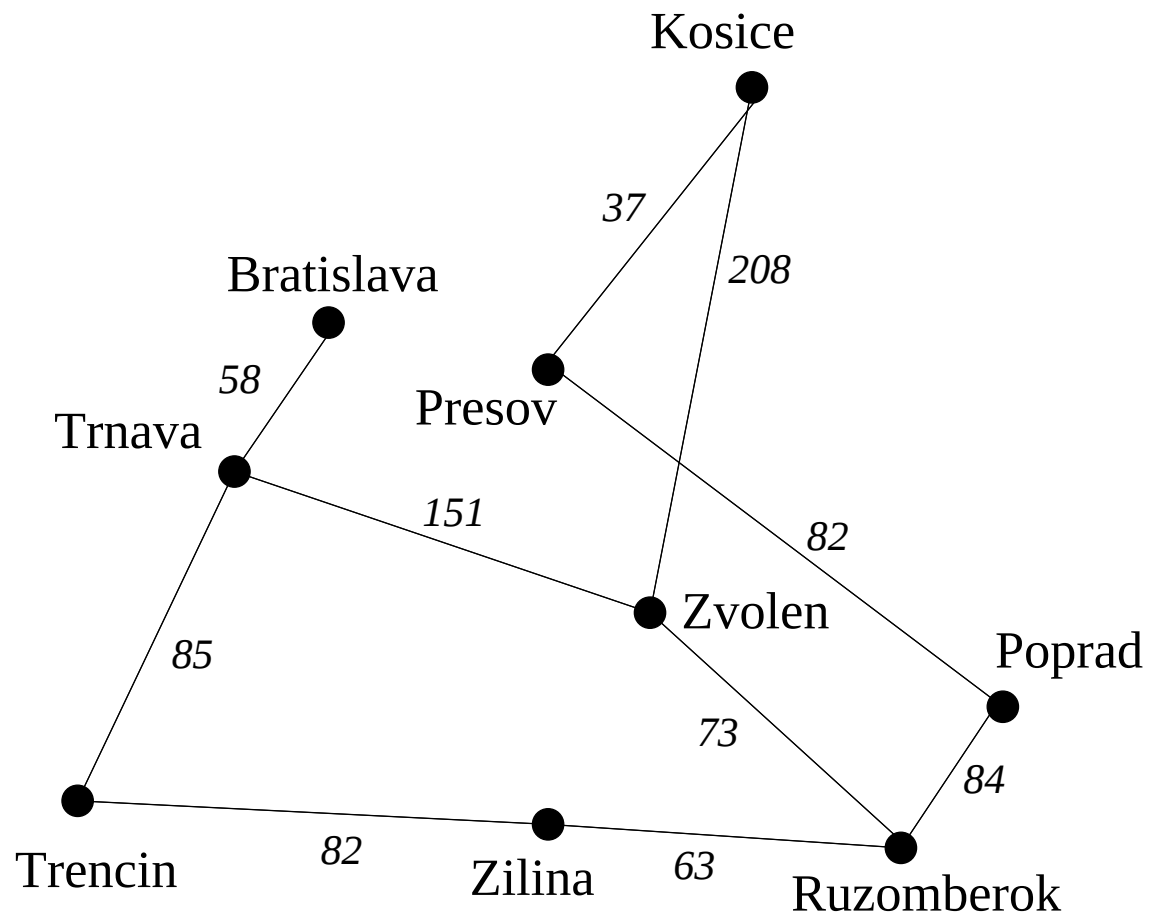
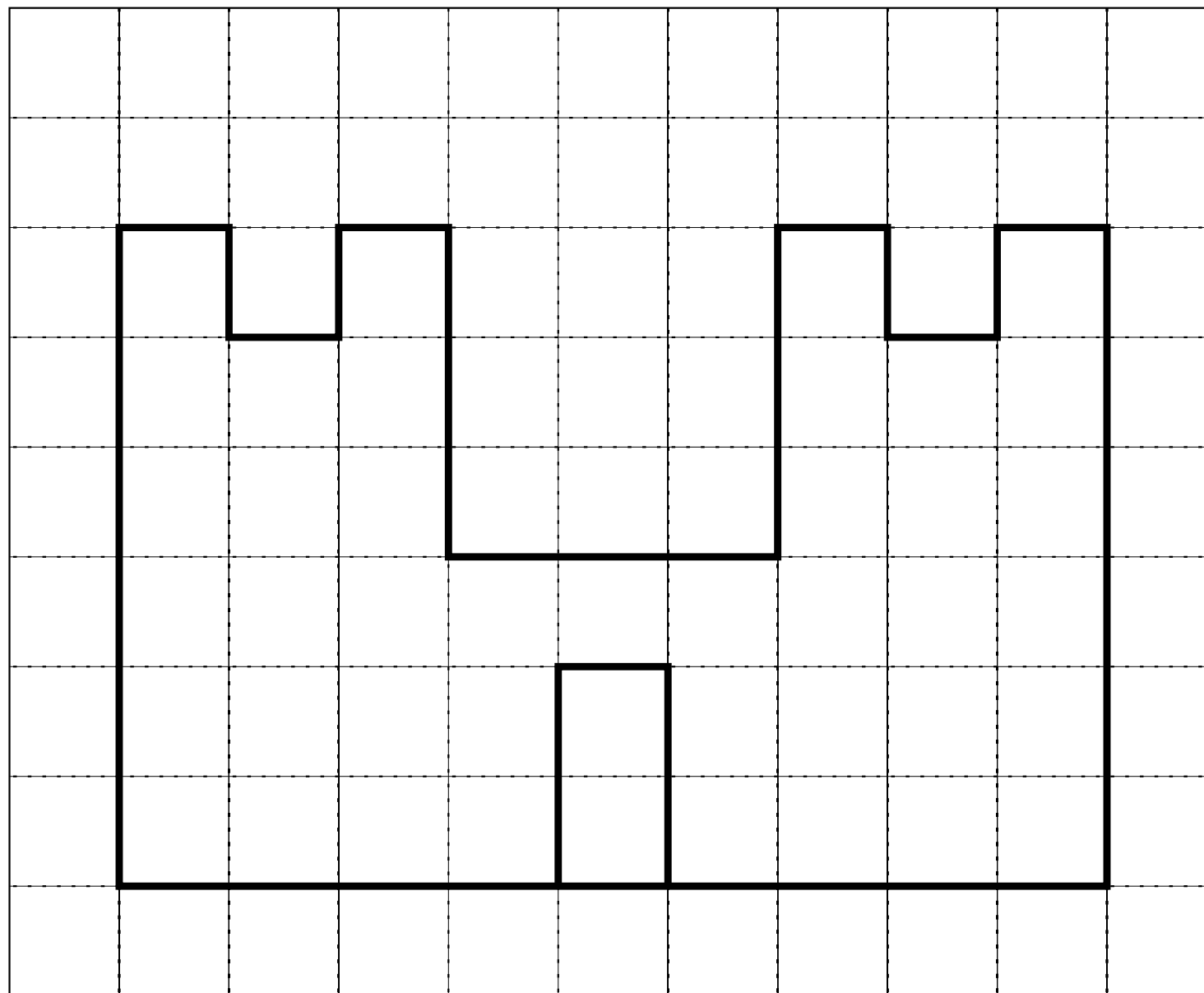
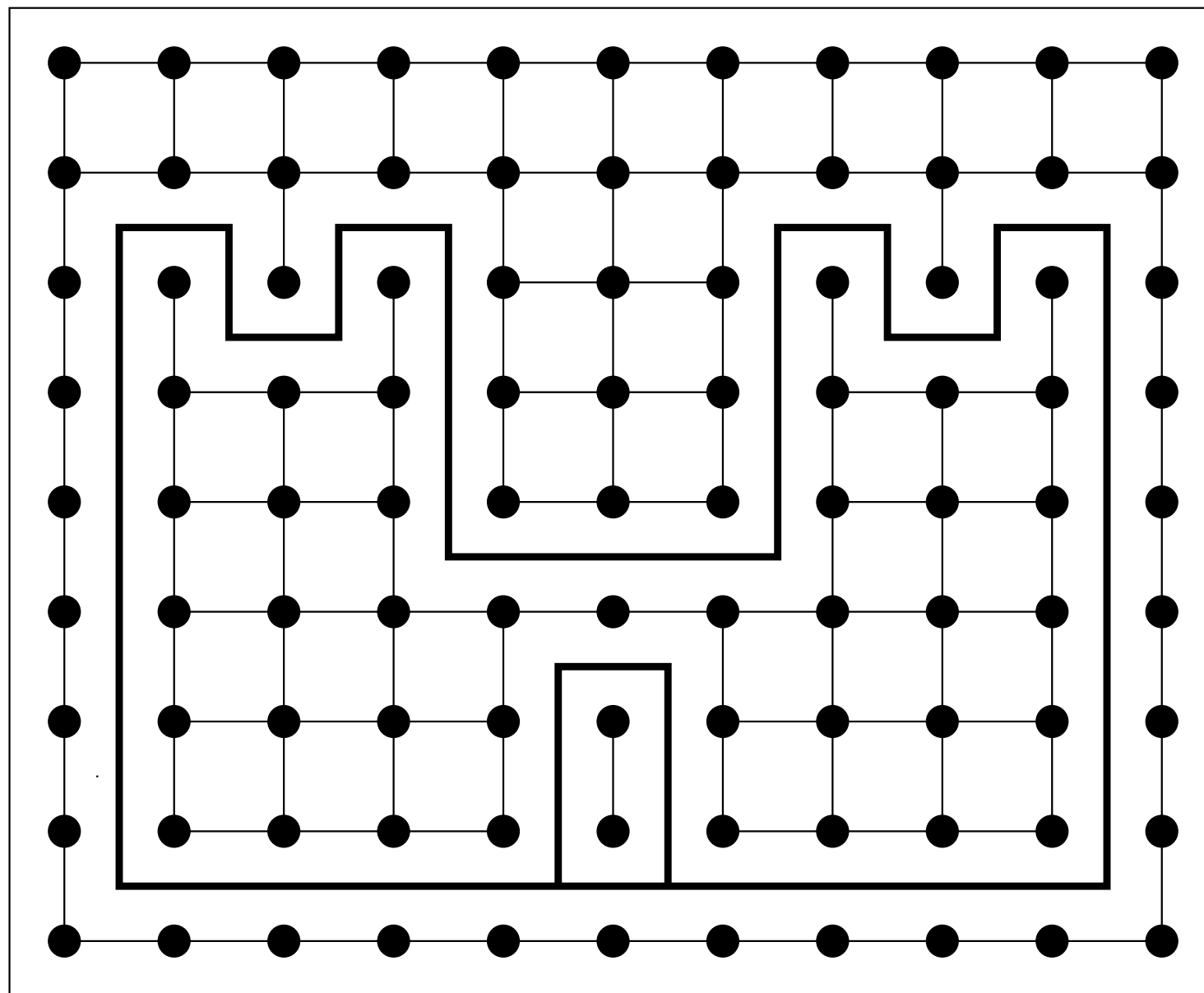










Prehľadávanie do hĺbky—rekurzívna funkcia

```
function dfs-visit(v,cnum)
    // pre-condition: v is WHITE vertex
    // find all vertices that are reachable from v
    // by path going through white vertices only
    status[v]:=gray;
    num[v]:=cnum;
    for each w in out(v)
        if status[w]=white
            dfs-visit(w,cnum)
    status[v]:=black;
```

Prehľadávanie do hĺbky—hlavný program

```
// --- main program ---  
status of all vertices is white  
cnum=0; // component number  
for all vertices v in V  
    if status[v]=white  
        // all vertices in v's component are white;  
        // explore v's component  
        dfs-visit(v,cnum);  
        cnum:=cnum+1;
```

Prehľadávanie do hĺbky—pamätáme si viac

```
function dfs-visit(v, cnum)
    status[v] := gray;
    * time := time + 1; d[u] := time;
    num[v] := cnum;
    for each w in out(v)
        if status[w] = white
            * edge (v, w) is a tree edge;
              dfs-visit(w, cnum)
    status[v] := black;
    * time := time + 1; f[u] := time;
```

Artikulácie: modifikácia prehľadávanie do hĺbky (vrchol, ktorý nie je koreň)

```
function dfs-visit(v, cnum)
    status[v] := gray;
    time := time + 1; d[v] := time;
*   detour[v] := v;
    for each w in out(v)
        if status[w] = white
            //--- (v,w) is a TREE edge
            dfs-visit(w, cnum); // detour[w] is now computed!
*       if detour[w] >= d[v] then vertex v is an articulation!
*       if detour[w] < detour[v] then detour[v] := detour[w];
*   else
*       //--- (v,w) is a BACK edge
*       if d[w] < detour[v] then detour[v] := d[w];
    status[v] := black;
```

Prehľadávanie do šírky

```
function bfs-visit(v,cnum)
    create empty queue Q;
    status[v]:=gray;
    dist[v]:=0;
    enqueue(Q,v);

    while Q is not empty
        u:=dequeue(Q);
        num[u]:=cnum;
        for each w in out(u)
            if status[w]=white then
                status[w]:=gray;
                dist[w]:=dist[u]+1;
                enqueue(Q,w);
        status[u]:=black;
```