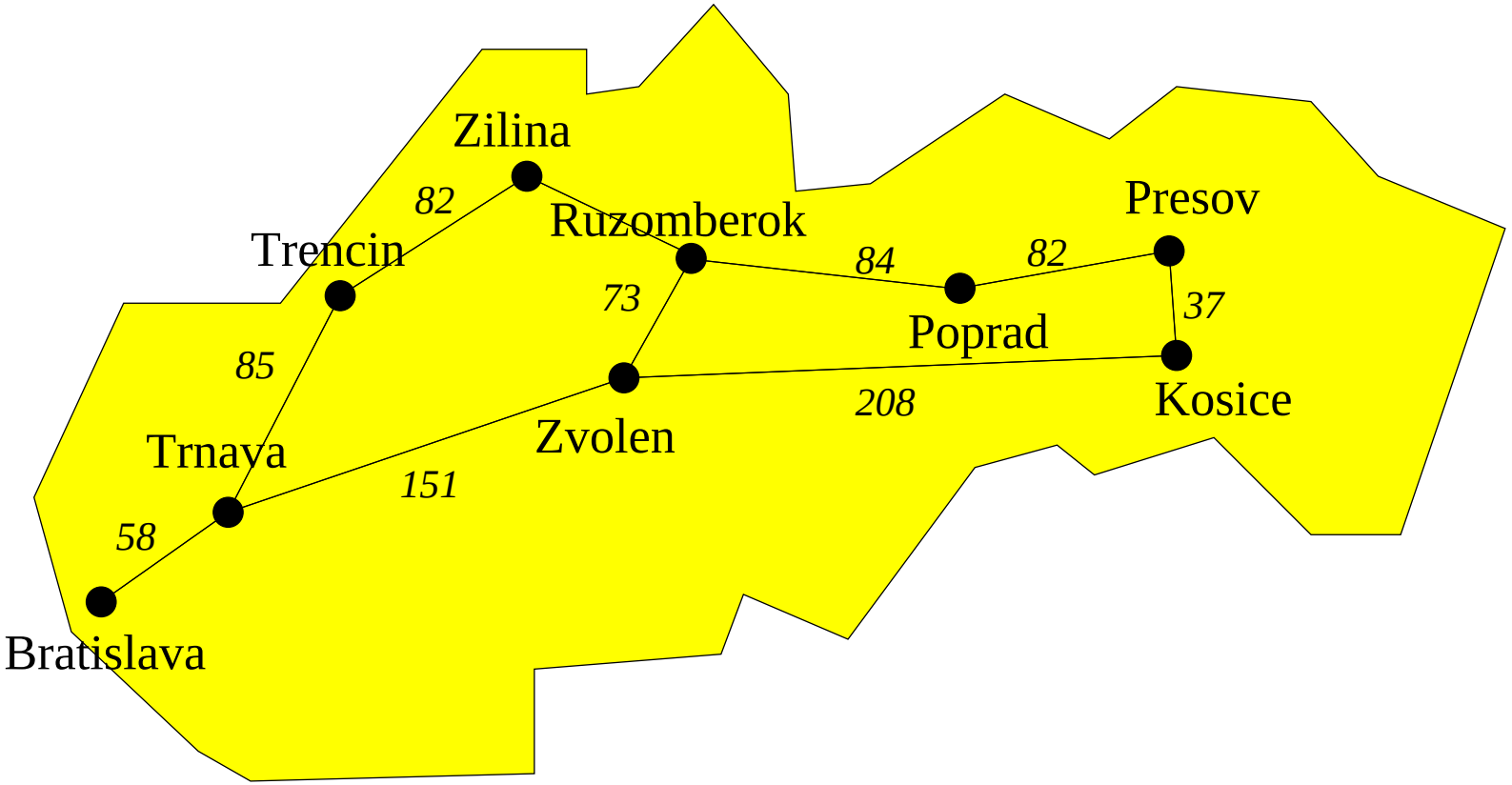
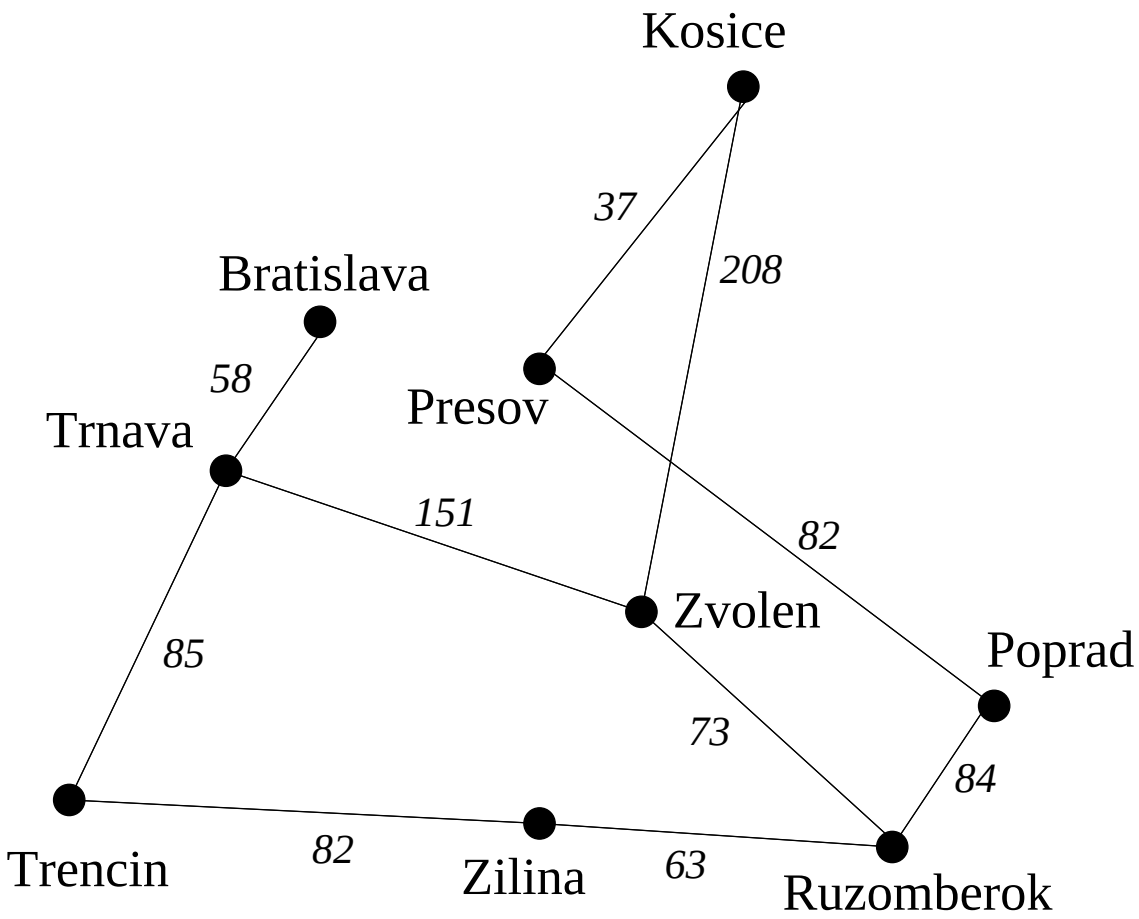

Zilina
82
Ruzomberok
Presov
Trencin
84
82
73
37
Poprad
85
208
Kosice
Zvolen
Trnava
151
58
Bratislava


Kosice
37
208
Bratislava
58
Presov
Trnava
151
82
Zvolen
85
Poprad
73
84
Trencin
82
Zilina
63
Ruzomberok

Dijkstrov algoritmus

```
// initialize dist, S, T
S:=0; T:=V;
for all w in V do dist[w]:=infinity;
dist[u]:=0;

// add one vertex at a time to T
while T is non-empty do
**s:=vertex for which dist[s] represent the length
**   of the shortest path from u to s;
  add s to S; remove s from T;
  // update dist to account for enlarged set S
  for all t in out(s) do
    // try to shorten current path to t through s
    if (dist[s]+w[s,t]<dist[t]) then
      dist[t]:=dist[s]+w[s,t];
```

Dijkstrov algoritmus

```
// initialize dist, S, T
S:=0; T:=V;
for all w in V do dist[w]:=infinity;
dist[u]:=0;

// add one vertex at a time to T
while T is non-empty do
  s:=vertex with the smallest dist[s];
  add s to S; remove s from T;
  // update dist to account for enlarged set S
  for all t in out(s) do
    // try to shorten current path to t through s
    if (dist[s]+w[s,t]<dist[t]) then
      dist[t]:=dist[s]+w[s,t];
```

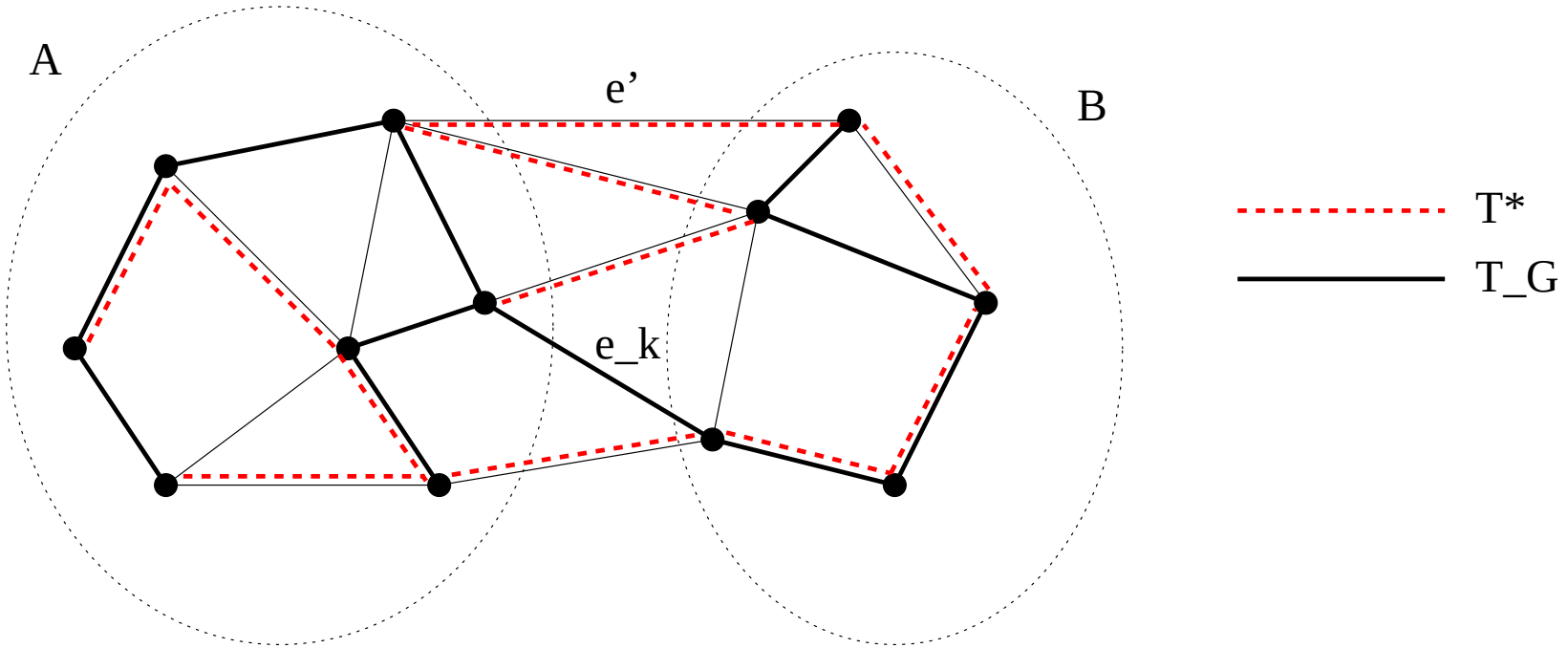
Dijkstrov algoritmus (s rekonštrukciou ciest)

```
// initialize dist, S, T
S:=0; T:=V;
for all w in V do
* dist[w]:=infinity; last[w]:=undefined;
dist[u]:=0;

// add one vertex at a time to T
while T is non-empty do
  s:=vertex with the smallest dist[s];
  add s to S; remove s from T;
  // update dist to account for enlarged set S
  for all t in out(s) do
    // try to shorten current path to t through s
    if (dist[s]+w[s,t]<dist[t]) then
*     dist[t]:=dist[s]+w[s,t]; last[t]:=s;
```

```
// path reconstruction from u to v
w:=v; create an empty path;
while last[w]<>undefined do
  add w to the beginning of the path;
  w:=last[w];
```

Kruskalov algoritmus (minimálna kostra)

```
Sort edges in order of increasing weight
so that w[f[1]] <= w[f[2]] <= ... <=w[f[m]]

T=empty set
for i:=1 to m do
  let u,v be the endpoints of edge f[i]
  if there is no path between u and v in T then (**)
    add f[i] to T
return T
```


A
e’
B
e_k
T*
T_G