

Naloga za narest:

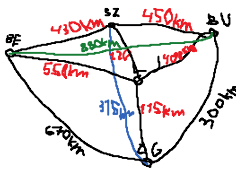
Denimo, da je n mest povezanih med seboj s poleti po principu vsak z vsakim. Iz prvega mesta želimo obiskati vsa ostala mesta tako, da nobenega mesta ne obiščemo več kot enkrat (pri tem je vseeno, v katerem mestu končamo). Dolžine poletov med posameznimi mesti so znane, iščemo pa najkrajšo pot (skupno dolžino poletov) iz prvega mesta, ki obišče vsa ostala mesta.

- a) Definirajte konkretno nalogo velikosti 5 in jo ustrezno predstavite.
b) Napišite prevdokodo za nek algoritem, ki rešuje ta problem.
c) Napišite sled algoritma za reševanje naloge a).

a) Definicija problema

Med seboj so povezana mesta, Ljubljana, Zagreb, Budimpešta, Salzburg, Bern po principu **vsak z vsakim**, vsa mesta bi si radi **ogledali le enkrat**, začeli bi v Ljubljani, ni pa važno v katerem mestu končamo naš ogled, za potovanje od enega mesta do drugega pa bi porabili najmanj kilometrov kot jih je možno porabiti

LJ = Ljubljana
ZG = Zagreb
BU = Budimpešta
SZ = Salzburg
BE = Bern



LJ -- ZG = 115km	ZG -- LJ = 115km
LJ -- BU = 400km	BU -- LJ = 400km
LJ -- SZ = 220km	SZ -- LJ = 220km
LJ -- BE = 550km	BE -- LJ = 550km
ZG -- BU = 300km	BU -- ZG = 300km
ZG -- SZ = 115km	SZ -- ZG = 115km
ZG -- BE = 670km	BE -- ZG = 670km
BU -- SZ = 315km	SZ -- BU = 315km
BU -- BE = 880km	BE -- BU = 880km
SZ -- BE = 430km	BE -- SZ = 430km

BE -- LJ = 550km	SZ -- LJ = 220km
BE -- ZG = 670km	SZ -- BE = 430km
BE -- SZ = 430km	SZ -- BU = 450km
BE -- BU = 880km	SZ -- ZG = 315km

BU -- LJ = 400km
BU -- ZG = 450km
BU -- SZ = 300km
BU -- BE = 880km

Algoritm:

To nalogo sem si razdelil na 2 problema,
prvi: generiraj pot in si jo zapomni (shrani)
drugi: med novo generirano potjo in shranjeno potjo poišči najbližjo, ter si jo na novo zapomni

generiraj pot in si jo zapomni (shrani)
nekje začneš poiščeš vse sosedje na tej točki, nato se premakneš na prvega sosedja, si zabeležiš pot in zabeležiš sosedja da si ga že obiskal
spet poiščeš vse sosedje, če je sosed že obiskal ga pustiš pri miru, drugače ga obiščeš, to delaš toliko časa dokler je še kaj ne obiskanih sosedov, vmes pa seštevaš razdalje od sosedja do sosedja

```
public class Pot()
{
    public String DefiniranaPot;
    public int DolzinaPoti;
};

public Pot StaraPot = null;
public Pot GenerirajPot()
{
    Pot novaPot = null;
    /* S magijo iskanja sosedov generiraj pot ter seštej razdalje*/

    //novaPot = {"LJ -- BE -- ZG -- BU -- SZ" : 1970}; //(DefiniranaPot : DolzinaPoti)

    StaraPot = poiisciNajblizjoPot(novaPot);

    return StaraPot;
}
```

med novo generirano potjo in shranjeno potjo poišči najbližjo, ter si jo na novo zapomni (funkcija naj vrne najbližjo pot)

```
public Pot poiisciNajblizjoPot(Pot NovaPot)
{
    If (StaraPot = null) //Če imamo definirano šele prvo pot (prvič kličemo funkcijo)
    {
        return NovaPot;
    }
    /* {"LJ -- BE -- ZG -- BU -- SZ" : 1970} <= {"LJ -- SZ -- BU -- ZG -- BE : 1640}
    If (NovaPot.DolzinaPoti <= StaraPot.DolzinaPoti) //Če je nova pot krajša kot prejšnja pot
    {
        return NovaPot; //{"LJ -- SZ -- BU -- ZG -- BE : 1640}
    }
}
```

b) Simulacija algoritma:

določiš prvo pot, določiš njeno razdaljo in si jo zapomniš kot najkrajšo v prvi vrstici

nato določiš drugo pot in njeno skupno razdaljo - če je ta manjša kot doslej najkrajša, si jo zapomniš

in tako naprej pregledaš vse poti do konca

Začnemo v točki LJ

LJ -- ZG -- BU -- SZ -- BE = 115 + 300 + 450 + 430 = 1.295 km
LJ -- ZG -- BE -- SZ -- BU = 115 + 670 + 430 + 450 = 1.665 km
LJ -- BU -- ZG -- BE -- SZ = 400 + 300 + 670 + 430 = 1.800 km
LJ -- BU -- SZ -- BE -- ZG = 400 + 450 + 430 + 670 = 1.950 km
LJ -- BE -- SZ -- BU -- ZG = 550 + 430 + 450 + 300 = 1.730 km
LJ -- BE -- ZG -- BU -- SZ = 550 + 670 + 300 + 450 = 1.970 km
LJ -- SZ -- BU -- ZG -- BE = 220 + 450 + 300 + 670 = 1.640 km

Najbližja pot: LJ -- ZG -- BU -- SZ -- BE = 115 + 300 + 450 + 430 = 1.295 km