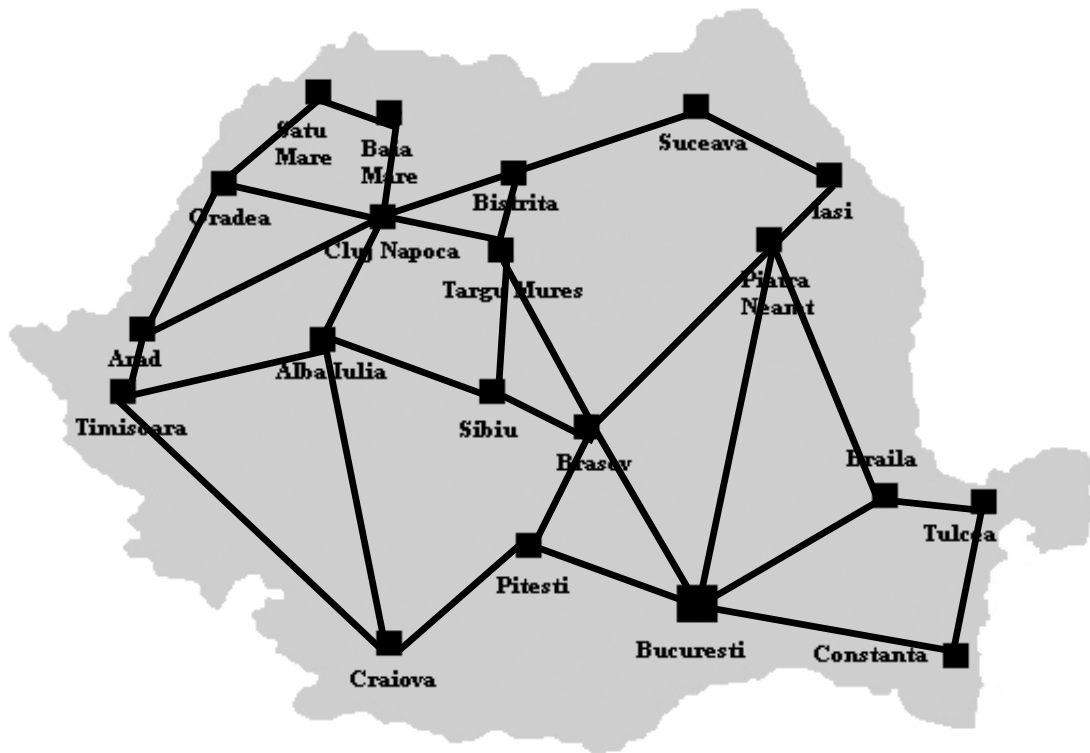


# Căutare cu cost uniform



Găsiți distanțele rutiere dintre orașele de pe harta din figură. Utilizați-le apoi pentru a implementa un algoritm de căutare cu cost uniform (folosind instrucțiunile de mai jos) pentru a ajunge de la un oraș la altul, de la Oradea la Tulcea.

## Instrucțiuni

### Algoritm de căutare cu cost uniform

Toate orașele sunt nevizitate.

Adăugăm în lista *noduri* orașul de plecare.

Marcăm orașul de plecare ca vizitat.

Inițializăm costul orașului de plecare cu 0.

*Cat timp* soluție negăsită și *noduri*  $\neq \emptyset$  *executa*

*nod* = *scoate\_din\_față*(*noduri*) //stocăm primul element din *noduri* în variabila *nod*

Eliminăm primul element din *noduri*

*Dacă* *testare\_țintă*[*problema*] se aplica la *stare*(*nod*) *atunci*

Soluția este găsită //facem variabila booleană *găsit* adevărată

*Altfel*

Găsim orașele conectate de *nod* și calculăm pentru ele costul ca fiind suma dintre costul lui *nod* + distanța de la *nod* la ele

Identificăm (orașele nevizitate care sunt conectate de *nod*) și (orașele vizitate anterior dacă au un cost mai mic decât aveau când au fost vizitate anterior)

Stabilim costul pentru orașe găsite (suma dintre costul lui *nod* + distanța de la *nod* la ele)

Adăugăm aceste orașe la *noduri* astfel încât orașele din *noduri* sunt ordonate după costul fiecăreia

Orașele adăugate sunt marcate ca vizitate

Se reține pentru oricare din orașele adăugate nodul *părinte* ca fiind *nod*

*Sfârșit cat timp*

Stocăm soluția parcurgând orașele de la destinație către start utilizând *părinții* reținuți.

Afișăm soluția și costul (numărul de km al soluției).