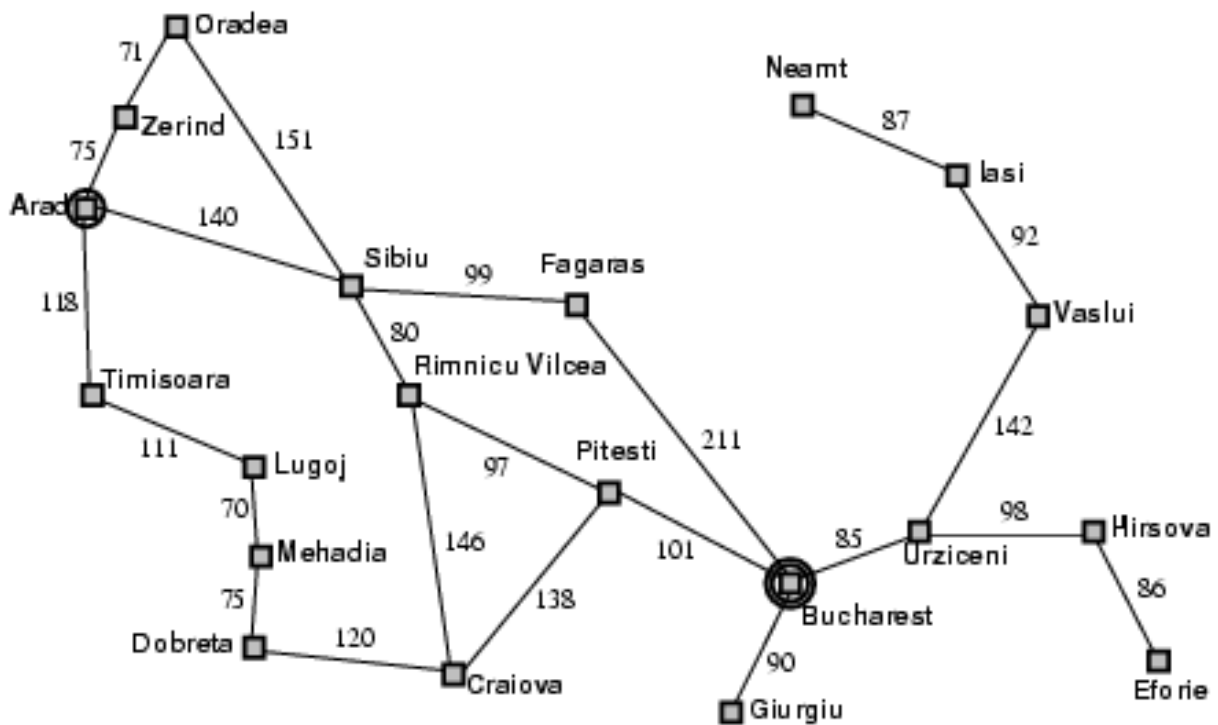


Fie harta de mai jos:



1. Implementați algoritmul de căutare în adâncime (adăugarea se face la început în lista *noduri* spre deosebire de căutarea în lățime).

Algoritm de căutare în adâncime

Toate orasele sunt nevizitate.

Adaugam în lista *noduri* orasul de plecare.

Marcam orasul de plecare ca vizitat.

Cat timp solutie negasita si *noduri* $\neq \emptyset$ *executa*

nod = scoate_din_fata(*noduri*) //stocam primul element din *noduri* in variabila *nod*

Eliminam primul element din *noduri*

Daca testare_tinta[problema] se aplica la stare(*nod*) *atunci*

Solutia este gasita //facem variabila booleana gasit adevarata

Altfel

Adaugam la început în *noduri* orasele nevizitate care sunt conectate de *nod*

Orasele adaugate sunt marcate ca vizitate

Se retine pentru oricare din orasele adaugate nodul *parinte* ca fiind *nod*

Sfarsit cat timp

Stocam solutia parcurgand orasele de la destinatie catre start utilizand *parintii* retinuti.

Sugestie

Începeți pornind de la căutarea în lățime (faceți o copie funcției de căutare în lățime și o redenumiți adâncime) și modificați doar adăugarea nodului nou în *noduri*: în loc să îl puneți pe ultima poziție, va trebui să îl punem pe prima poziție prin mutarea tuturor elementelor la dreapta cu o poziție.