Presupunem, pentru simplificare, că vârfurile sunt numerotate, $V = \{1, 2, \ldots, n\}$, vârful 1 fiind sursa, şi că matricea L dă lungimea fiecărei muchii, cu $L[i, j] = +\infty$ dacă muchia (i, j) nu există. Soluţia se va construi în tabloul $D[2..n]$. Algoritmul este:

```
function DIJKSTRA(L[1..n, 1..n])
    {iniţializare}
    C ← {2, 3, ..., n}        {S = V \ C există doar implicit}
    for i ← 2 to n do
        D[i] ← L[1, i]
    end for
    {bucla greedy}
    repeat n − 2 times
        v ← vârful din C care minimizează D[v]
        C ← C \ {v}        {şi, implicit, S ← S ∪ {v}}
        for fiecare w ∈ C do
            D[w] ← min(D[w], D[v] + L[v, w])
        end for
    return D
end function
```

Pentru graful din Figura 1, paşii algoritmului sunt prezentaţi în Tabelul 1.

Observăm că D nu se schimbă dacă mai efectuăm o iteraţie pentru a-l scoate şi pe $\{2\}$ din C. De aceea, bucla greedy se repetă doar de $n - 2$ ori.

Se poate demonstra următoarea proprietate:

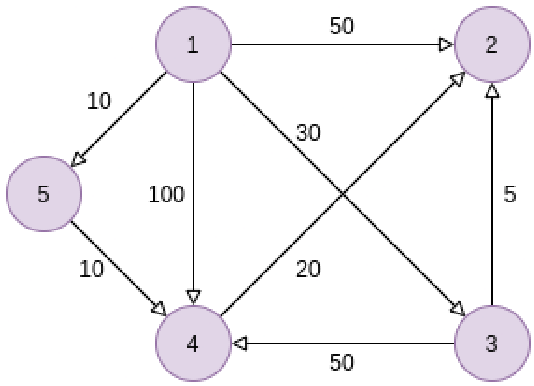

Figura 1: Un graf orientat

Pasul	v	C	D
iniţializare		$\{2, 3, 4, 5\}$	$[50, 30, 100, 10]$
1	5	$\{2, 3, 4\}$	$[50, 30, 20, 10]$
2	4	$\{2, 3\}$	$[40, 30, 20, 10]$
3	3	$\{2\}$	$[35, 30, 20, 10]$

Tabelul 1: Algoritmul lui Dijkstra aplicat grafului din Figura 1