Bellman-Ford

Cayke Prudente e Igor Miranda

Características

- Encontra o menor caminho em dígrafos ponderados.
- Funciona com pesos negativos.

Bellman X Dijkstra

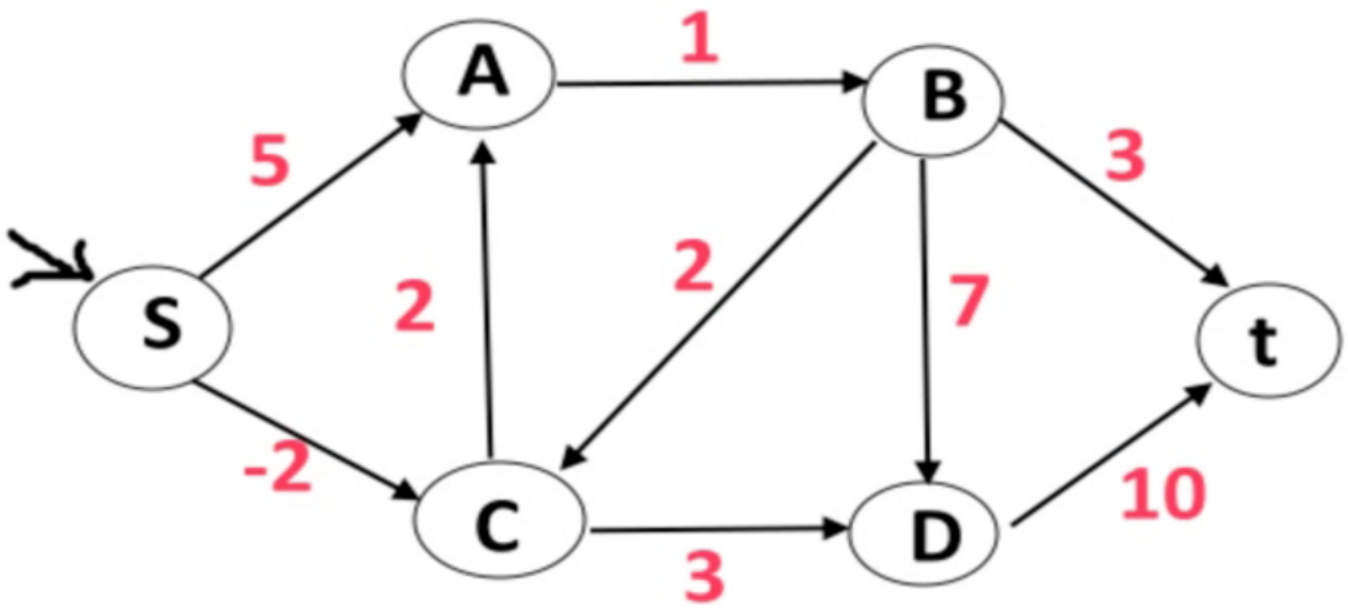
- Dijkstra é mais rápido, O(E + V.logV)
  - x O(V.E)
- Bellman é mais versátil.

Programação dinâmica

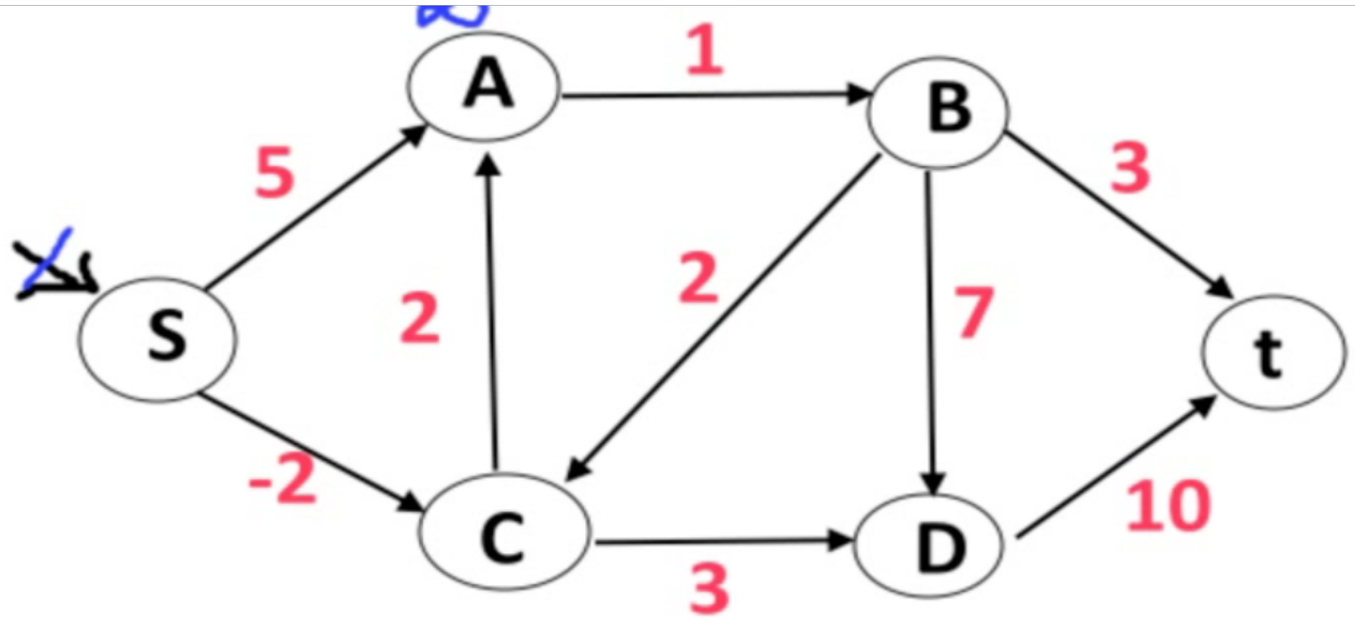
- Subestrutura ótima. A solução ótima para o problema contém em seu interior soluções ótimas para subproblemas.
- Superposição de subproblemas. O algoritmo reexamina o mesmo problema muitas vezes.

Funcionamento

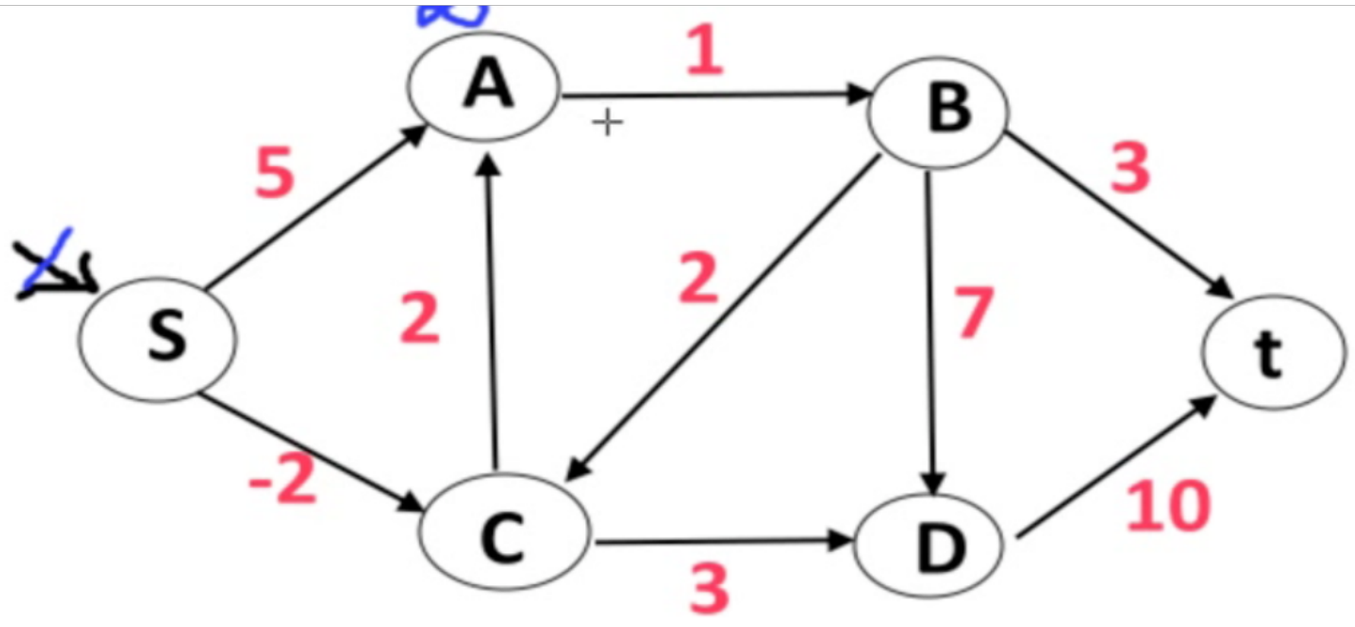
- Utiliza a técnica de relaxamento V-1 vezes.
  - Relaxamento: Para toda aresta (u,v), se distancia (s,u) + peso (u,v) < distancia (s,v), então temos um novo caminho.
  - Ao fim confere se não há ciclos negativos.



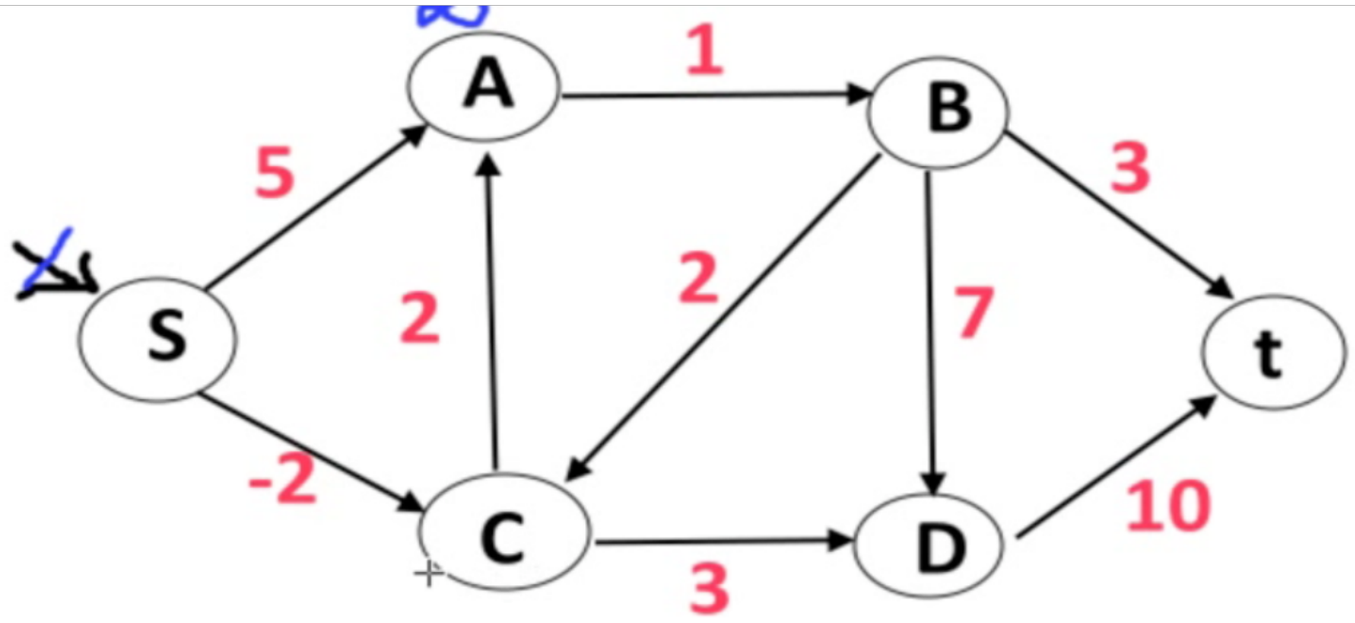
	S	A	B	C	D	t
d[v]	0	∞	∞	∞	∞	∞
Pi[v]						



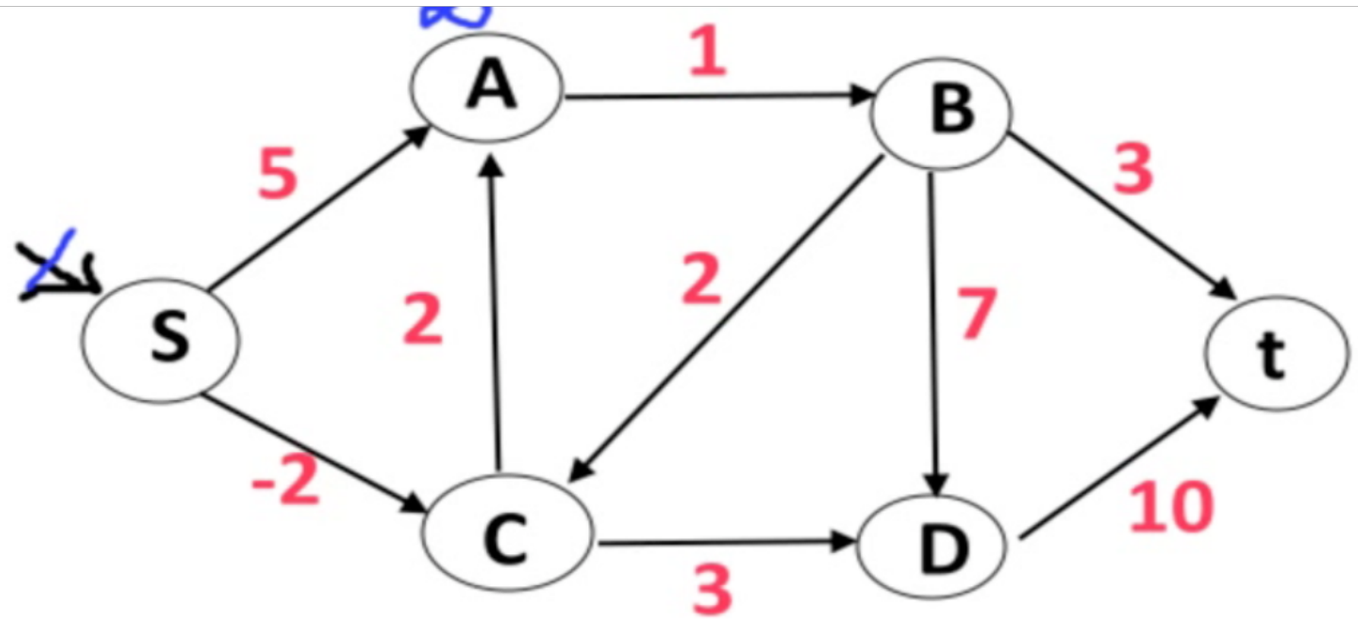
	S	A	B	C	D	t
d[v]	0	~~∞~~ 5	∞	~~∞~~ -2	∞	∞
Pi[v]		S		S		



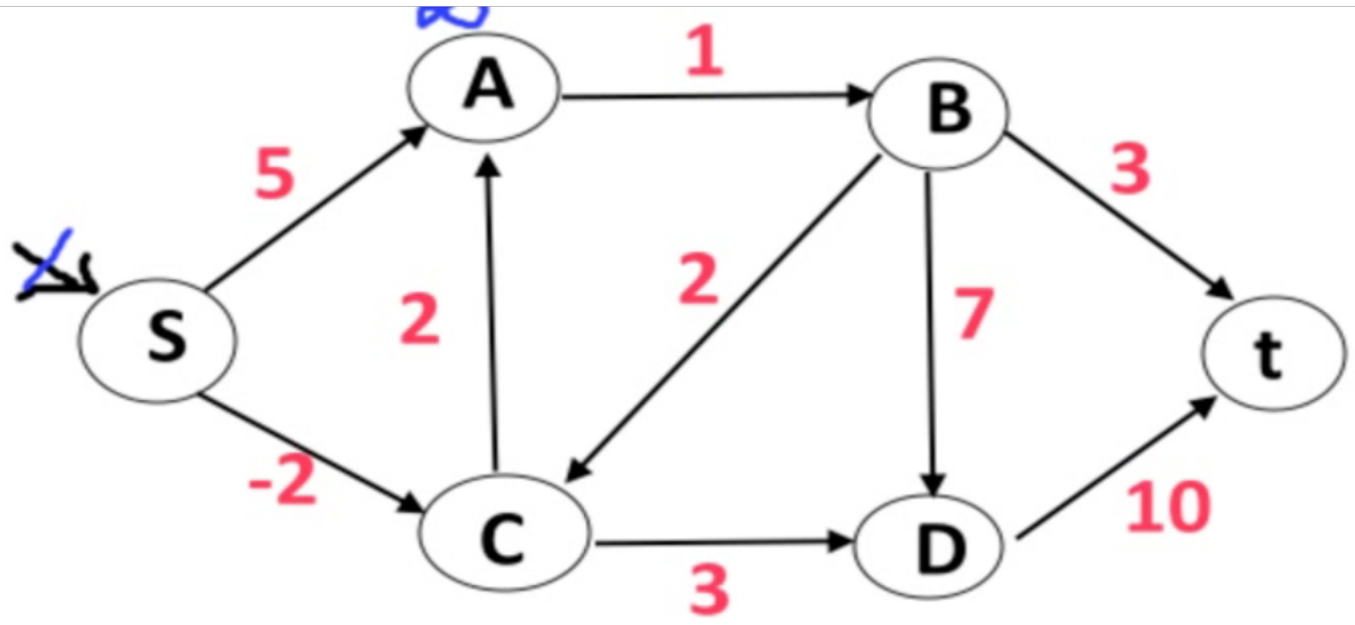
	S	A	B	C	D	t
d[v]	0	~~∞~~5	~~∞~~6	~~∞~~-2	∞	∞
Pi[v]		S	A	S		



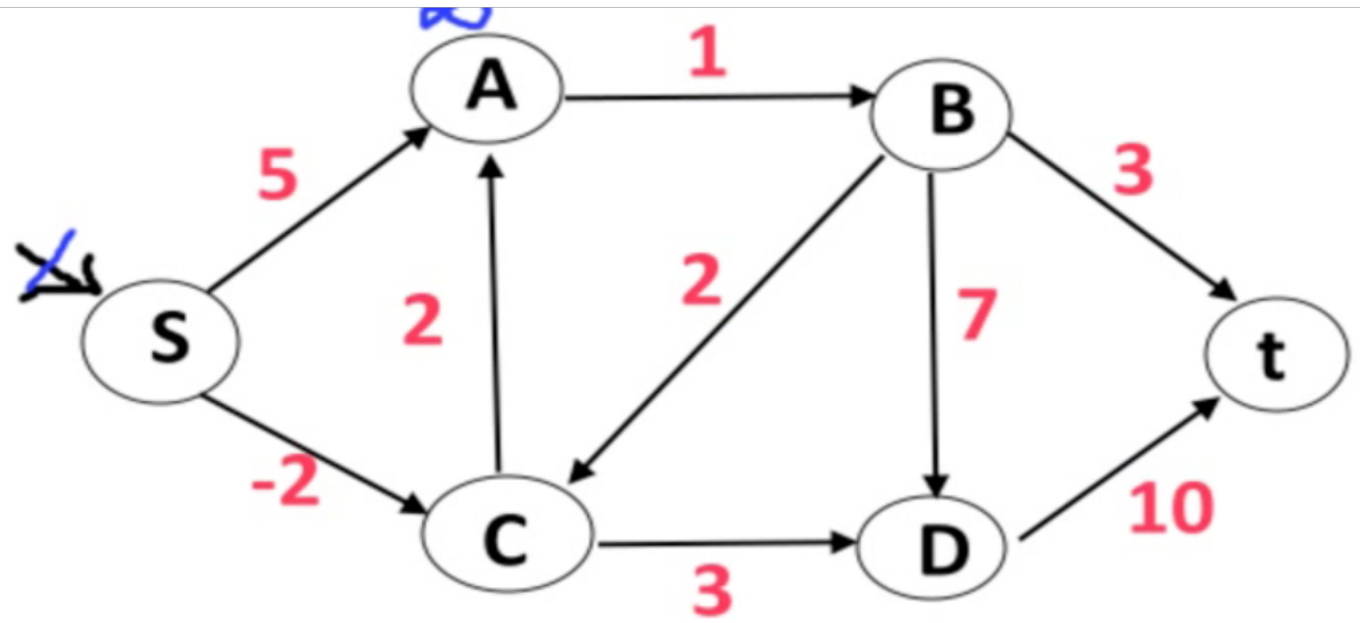
	S	A	B	C	D	t
d[v]	0	5	6	-2	13	9
Pi[v]		S	A	S	B	B



	S	A	B	C	D	t
d[v]	0	∞5∞0	∞6	∞-2	∞∞31	∞9
Pi[v]		∞C	A	S	BC	B



	S	A	B	C	D	t
d[v]	0	∞5∞0	∞6	∞-2	∞∞31	∞9
Pi[v]		$C	A	S	BC	B



	S	A	B	C	D	t
d[v]	0	∞5∞0	∞∞1	∞-2	∞∞31	∞∞4
Pi[v]	0	$C	#A	S	BC	BB


parameters.contains(
ql += " and p.name = :name";
if(parameters.contains("age")){
hql += " and p.age = :age";
TypedQuery<Person> query = em.createQuery(hql, Person
}
if(parameters.contains("name")){
query.setParameter("name", values[0].toString());
}
if(parameters.contains("age")){
meter("age", Integer.valueOf(values
list());