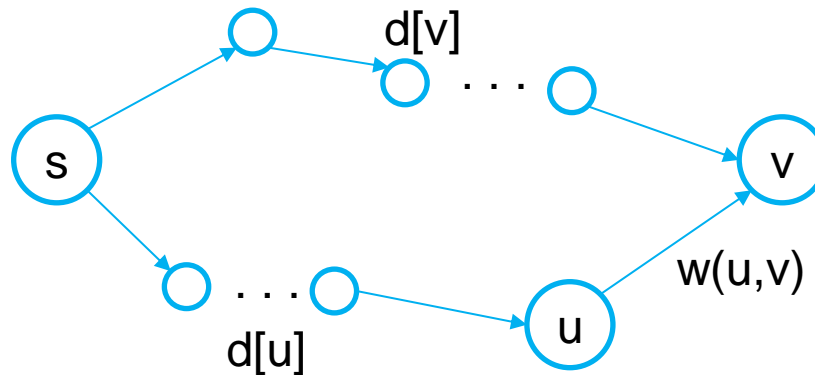


Thuật toán Dijkstra

- Cho đồ thị có hướng $G=(V, E, A, w)$
 - V : tập đỉnh
 - E : tập cung
 - A : cấu trúc danh sách kề, trong đó $A[v]$ là tập các cung kề với v (đi ra khỏi v)
 - $w(u, v)$: là trọng số (không âm) cung (u, v)
- Cho đỉnh $s \in V$, tìm đường đi ngắn nhất từ s đến tất cả các đỉnh còn lại của G

Thuật toán Dijkstra

- Cho đồ thị có hướng $G=(V, E, A, w)$
 - V : tập đỉnh
 - E : tập cung
 - A : cấu trúc danh sách kề, trong đó $A[v]$ là tập các đỉnh kề với v
 - $w(u,v)$: là trọng số (không âm) cung (u,v)
- Cho đỉnh $s \in V$, tìm đường đi ngắn nhất từ s đến tất cả các đỉnh còn lại của G
- Thuật toán Dijkstra
 - Duy trì $d[v]$: cận trên độ dài đường đi ngắn nhất từ s đến v
 - Khởi tạo: $d[v] = w(s,v)$ (giả thiết $w(s,v) = +\infty$, nếu $(s,v) \notin E$)
 - Mỗi khi phát hiện đỉnh u sao cho $d[v] > d[u] + w(u,v)$ thì làm tốt cận trên: $d[v] = d[u] + w(u,v)$



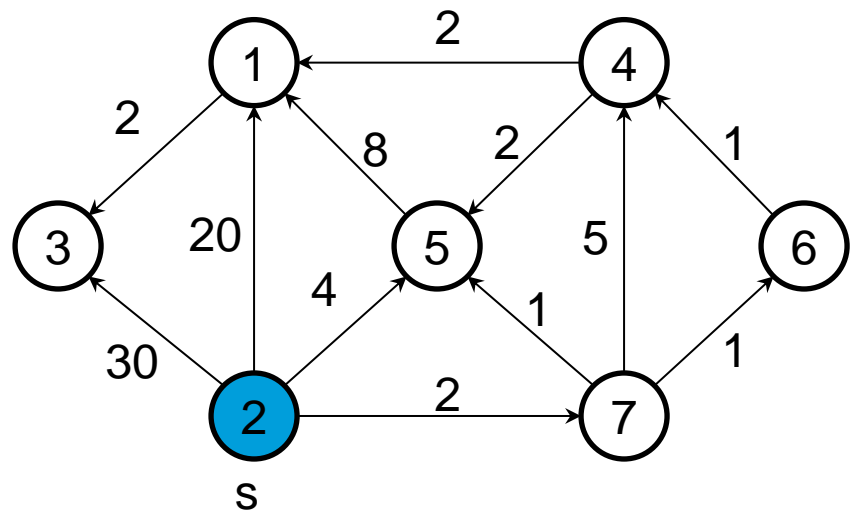
Thuật toán Dijkstra

- Thuật toán Dijkstra
 - Duy trì $d[v]$: cận trên độ dài đường đi ngắn nhất từ s đến v
 - Khởi tạo: $d[v] = w(s, v)$ (giả thiết $w(s, v) = +\infty$, nếu $(s, v) \notin E$)
 - Mỗi khi phát hiện đỉnh u sao cho $d[v] > d[u] + w(u, v)$ thì làm tốt cận trên:
$$d[v] = d[u] + w(u, v)$$

Đỉnh u được chọn sẽ là đỉnh mà cận trên $d[u]$ chính là độ dài đường đi ngắn nhất từ s đến u

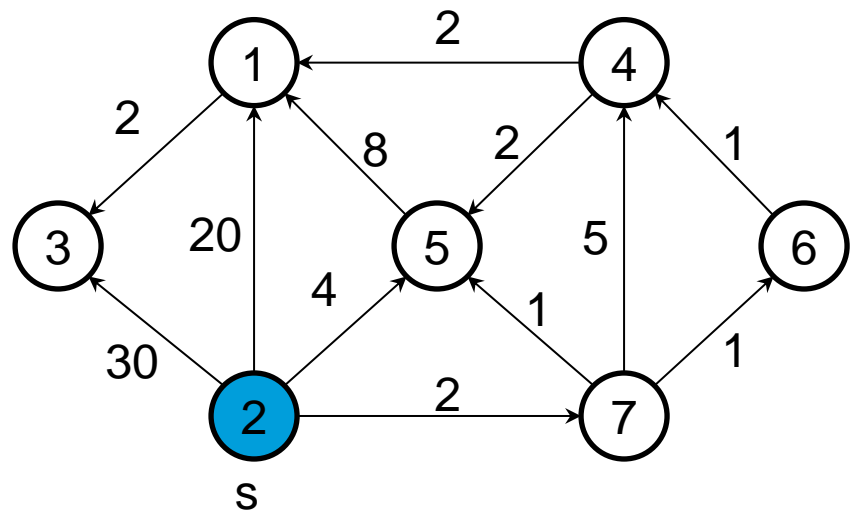
```
Dijkstra( $G = (V, E, w)$ ) {  
  for( $v \in V$ ) {  
     $d(v) = w(s, v)$ ;  
  }  
   $S = V \setminus \{s\}$ ; // tap cac dinh chua tim duoc duong  
    // di ngan nhac  
  while( $S \neq \{\}$ ) {  
     $u = \text{select a node } \in S \text{ having minimum } d(u)$ ;  
     $S = S \setminus \{u\}$ ;  
    for( $v \in S$ ) {  
      if( $d(v) > d(u) + w(u, v)$ ) {  
         $d(v) = d(u) + w(u, v)$ ;  
      }  
    }  
  }  
}
```

Thuật toán Dijkstra



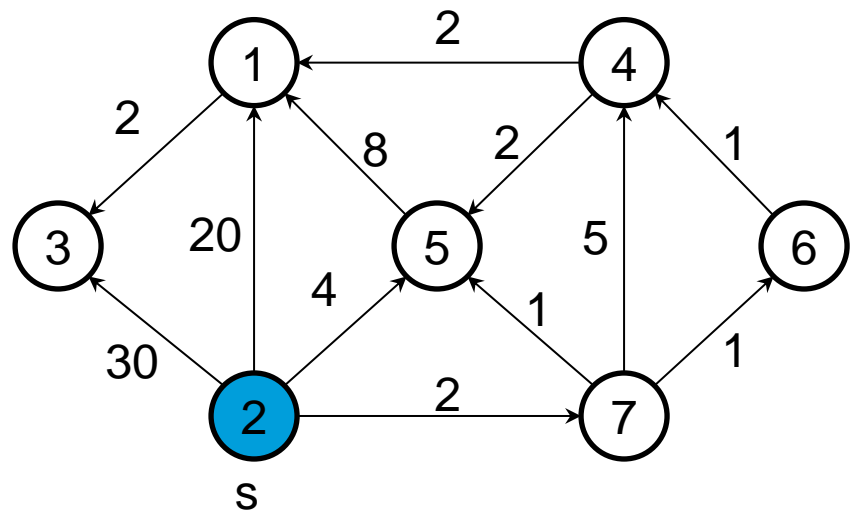
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1							
2							
3							
4							
5							
6							

Thuật toán Dijkstra



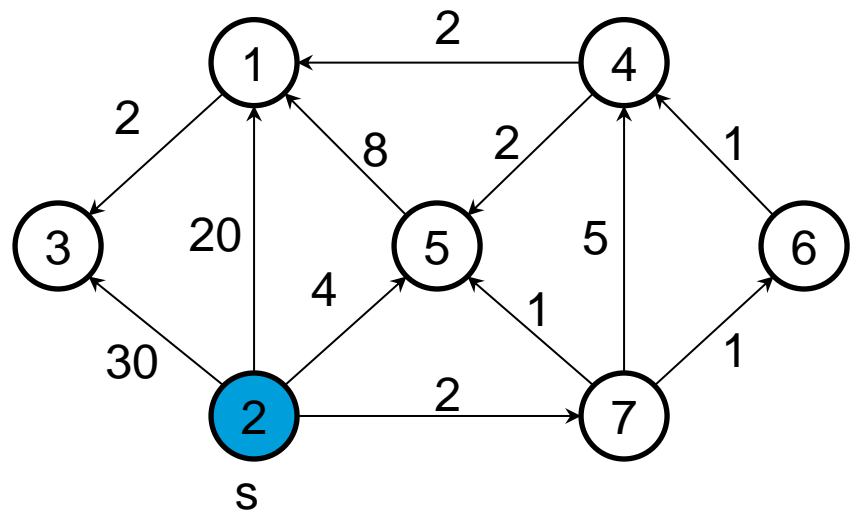
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1							
2							
3							
4							
5							
6							

Thuật toán Dijkstra



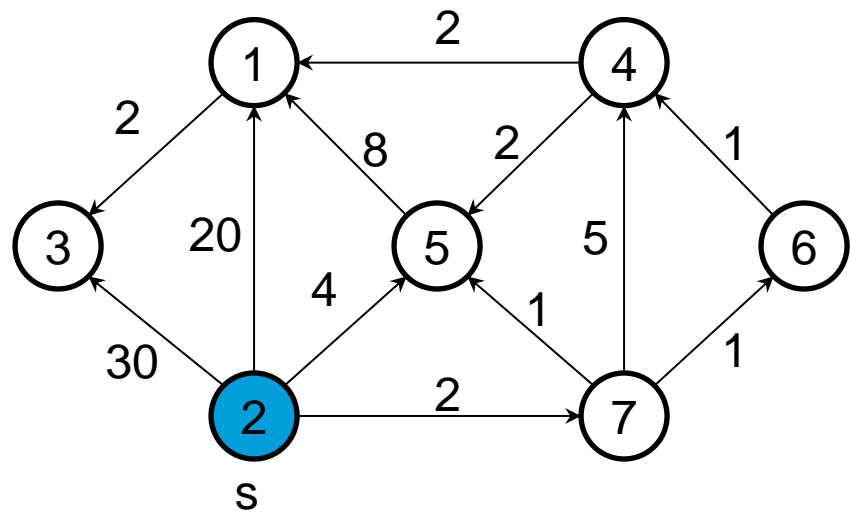
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2							
3							
4							
5							
6							

Thuật toán Dijkstra



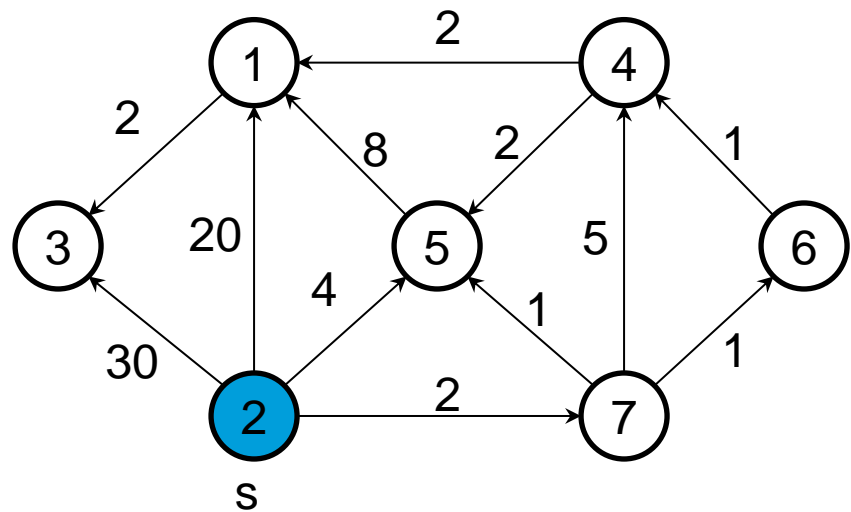
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2							
3							
4							
5							
6							

Thuật toán Dijkstra



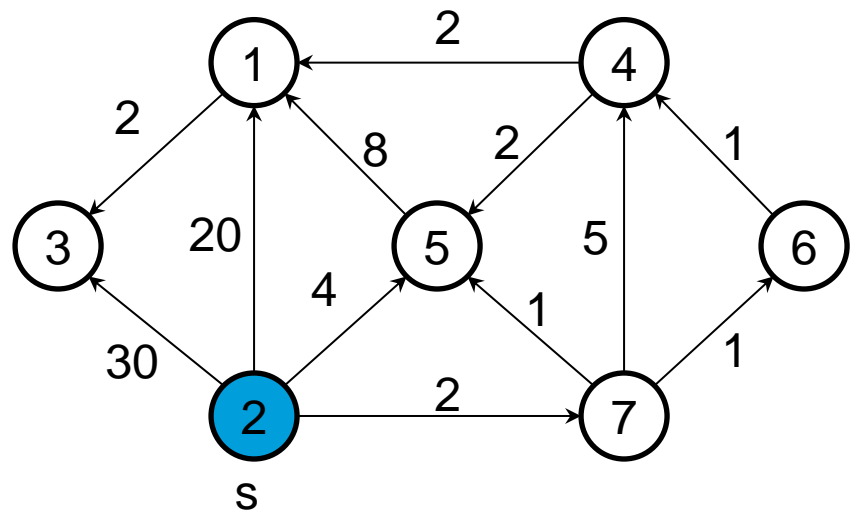
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3							
4							
5							
6							

Thuật toán Dijkstra



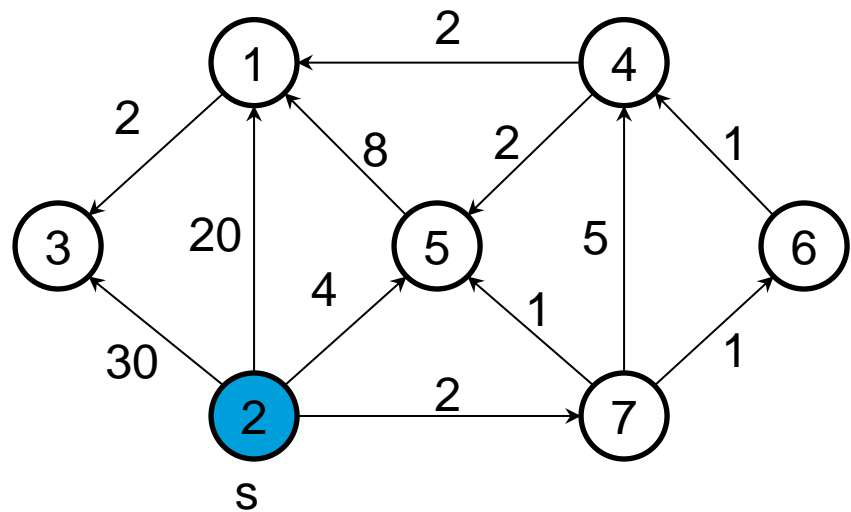
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3							
4							
5							
6							

Thuật toán Dijkstra



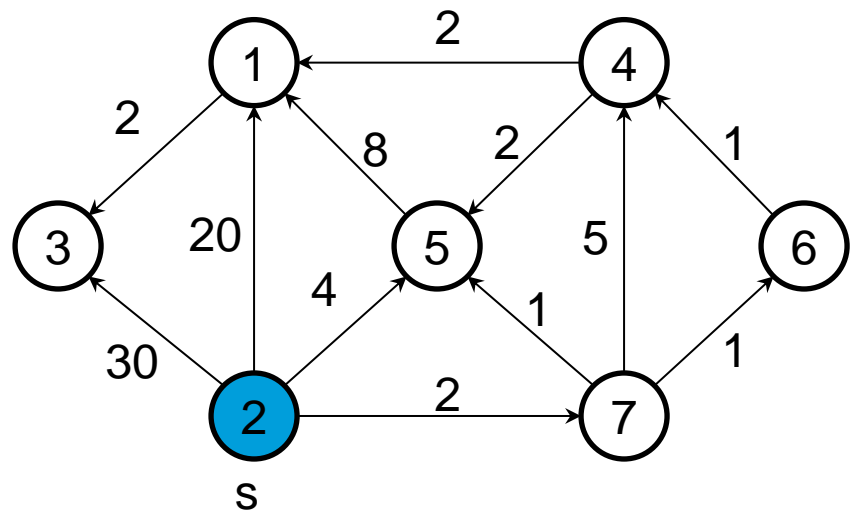
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4							
5							
6							

Thuật toán Dijkstra



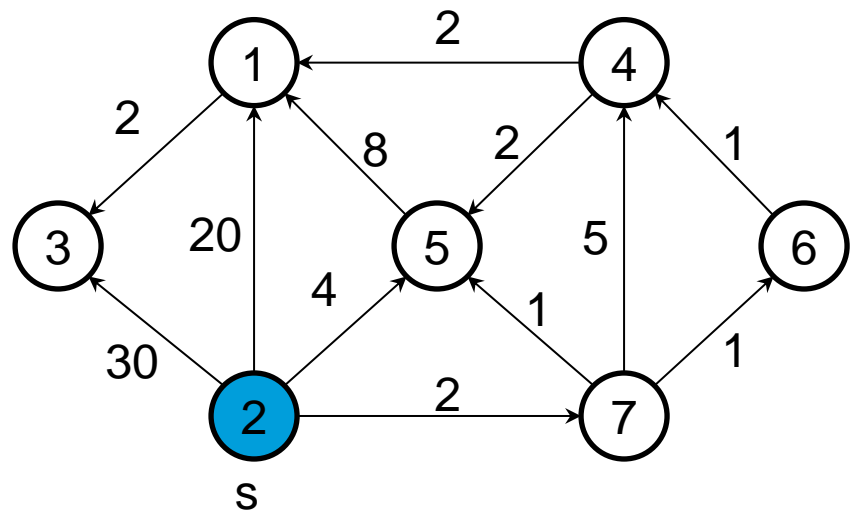
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4							
5							
6							

Thuật toán Dijkstra



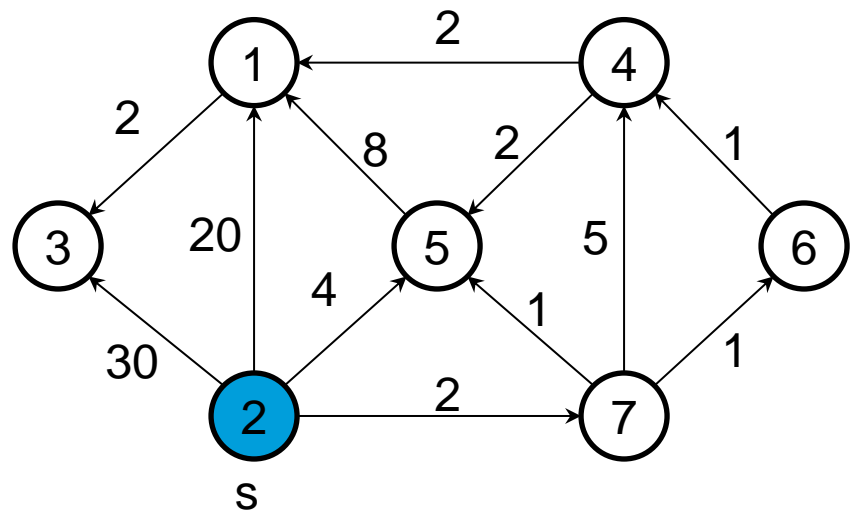
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4	6	0	30	4	3	3	2
5							
6							

Thuật toán Dijkstra



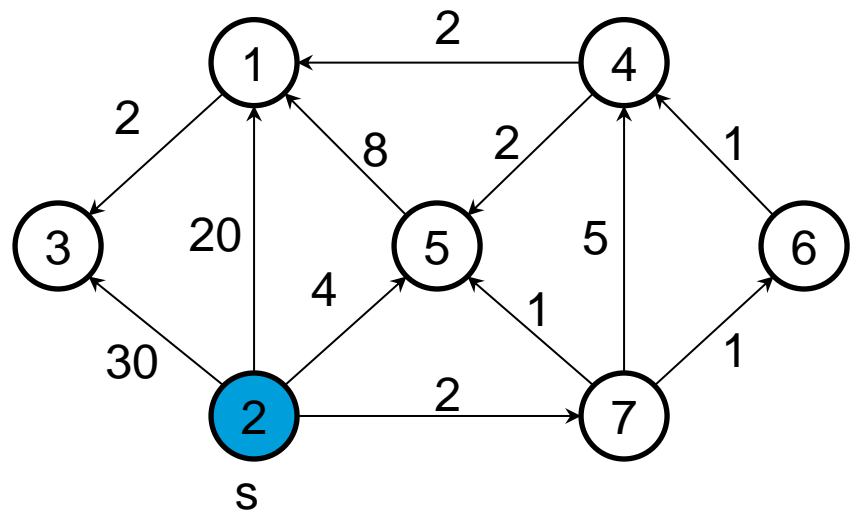
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4	6	0	30	4	3	3	2
5							
6							

Thuật toán Dijkstra



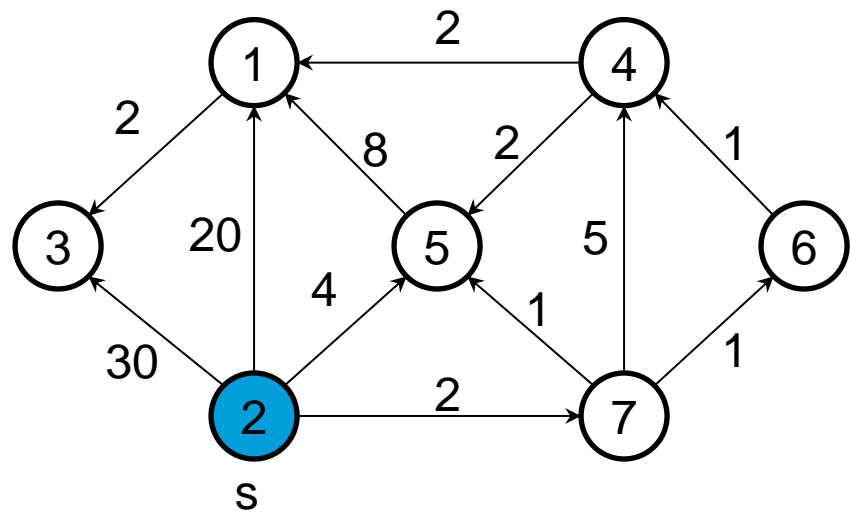
	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4	6	0	30	4	3	3	2
5	6	0	8	4	3	3	2
6							

Thuật toán Dijkstra



	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4	6	0	30	4	3	3	2
5	6	0	8	4	3	3	2
6							

Thuật toán Dijkstra



	1	2	3	4	5	6	7
Khởi tạo	20	0	30	$+\infty$	4	$+\infty$	2
1	20	0	30	7	3	3	2
2	20	0	30	4	3	3	2
3	11	0	30	4	3	3	2
4	6	0	30	4	3	3	2
5	6	0	8	4	3	3	2
6	6	0	8	4	3	3	2