

Cài đặt thuật toán Dijkstra sử dụng hàng đợi ưu tiên (Priority Queue)

- 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 2 đỉnh $s, t \in V$, tìm đường đi ngắn nhất từ s đến t trên 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 $(u,v) \notin E$)
 - Mỗi khi phát hiện đỉnh u sao cho $d[v] > d[u] + w(u,v)$ thì làm tốt cận trên:

$$d[v] = d[u] + w(u,v)$$

```
Dijkstra( $G = (V, E, w)$ ) {  
  for( $v \in V$ ) {  
     $d(v) = w(s, v)$ ;  
  }  
   $S = V \setminus \{s\}$ ; // tập các đỉnh chưa tìm được đường  
                   // đi ngắn nhất  
  while( $S \neq \{\}$ ) {  
     $u = \text{select a node } \in S \text{ having minimum } d(u)$ ;  
    if ( $u = t$ ) break; // found the shortest path  
     $S = S \setminus \{u\}$ ;  
    for( $v \in S$ ) {  
      if( $d(v) > d(u) + w(u, v)$ ) {  
         $d(v) = d(u) + w(u, v)$ ;  
      }  
    }  
  }  
}
```

Cài đặt thuật toán Dijkstra sử dụng hàng đợi ưu tiên (Priority Queue)

- Mã giả

```
1. DijkstraWithPriorityQueue(G = (V, A), s, t) {
2.   for v in V do d[v] =  $+\infty$ ;
3.   pq = initPQ(); pq.Push(s, 0);
4.   while(pq not empty){
5.     u = pq.Pop();
6.     for a in A[u] do {
7.       v = a.id; w = a.w;
8.       if d[v] > d[u] + w then pq.Push(v, d[u] + w);
9.     }
10.  }
11.  return d[t];
12. }
```