

1. BFS

	Queue	Visited	Parent (child-parent)
1	V1	V1	
2	V2 , V3, V4	V1, V2 , V3 , V4	V2-V1 , V3-V1 , V4-V1
3	V3 , V4, V7	V1, V2, V3, V4, V7	V2-V1, V3-V1, V4-V1, V7-V2
4	V4 , V7, V6, V8	V1, V2, V3, V4, V6 , V8	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3 , V8-V3
5	V7 , V6, V8, V5	V1, V2, V3, V4, V6, V8, V5	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4
6	V6 , V8, V5	V1, V2, V3, V4, V6, V8, V5	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4
7	V8 , V5, V12	V1, V2, V3, V4, V6, V8, V5, V12	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6
8	V5 , V12, V9, V10	V1, V2, V3, V4, V6, V8, V5, V12, V9 , V10	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8 , V10-V8
9	V12 , V9, V10, V14	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5
10	V9 , V10, V14, V13	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12
11	V10 , V14, V13, V11	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9
12	V14 , V13, V11	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9
13	V13 , V11	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9
14	V11	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9
15	V15	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11, V15	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9, V15-V11
16	V16	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11, V15, V16	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9, V15-V11, V16-V15
17	V17	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11, V15, V16, V17	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9, V15-V11, V16-V15, V17-V16
18	V18	V1, V2, V3, V4, V6, V8, V5, V12, V9, V10, V14, V13, V11, V15, V16, V17, V18	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V5-V4, V12-V6, V9-V8, V10-V8, V14-V5, V13-V12, V11-V9, V15-V11, V16-V15, V17-V16, V18-V17

Đường đi BFS : V1-V3-V8-V9-V11-V15-V16-V17-V18

2. DFS

	Queue	Visited	Parent (child-parent)
1	V1	V1	
2	V2, V3, V4	V1, V2, V3, V4	V2-V1,V3-V1,V4-V1
3	V2,V3, V5	V1, V2, V3, V4, V5	V2-V1,V3-V1,V4-V1, V5-V4
4	V2,V3, V14	V1,V2,V3,V4,V5, V14	V2-V1,V3-V1,V4-V1,V5-V4, V14-V5
5	V2,V3, V11	V1,V2,V3,V4,V5,V14, V11	V2-V1,V3-V1,V4-V1,V5-V4,V14-V5, V11-V14
6	V2,V3, V15	V1,V2,V3,V4,V5,V14,V11, V15	V2-V1,V3-V1,V4-V1,V5-V4,V14-V5,V11-V14, V15-V11
7	V2,V3, V16	V1,V2,V3,V4,V5,V14,V11,V15, V16	V2-V1,V3-V1,V4-V1,V5-V4,V14-V5,V11-V14,V15-V11, V16-V15
8	V2,V3, V17	V1,V2,V3,V4,V5,V14,V11,V15,V16, V17	V2-V1,V3-V1,V4-V1,V5-V4,V14-V5,V11-V14,V15-V11,V16-V15, V17-V16
9	V2,V3, V18	V1,V2,V3,V4,V5,V14,V11,V15,V16,V17, V18	V2-V1,V3-V1,V4-V1,V5-V4,V14-V5,V11-V14,V15-V11,V16-V15,V17-V16, V18-V17

Đường đi DFS : V1-V4-V5-V14-V11-V15-V16-V17-V18

3. UCS

	Priority Queue	Visited	Parent (child-parent)
1	{{ (V1,0) }}	V1	
2	{{ (V2,50) },(V3,350),(V4,300)}	V1, V2, V3, V4	V2-V1,V3-V1,V4-V1
3	{{(V7,650),(V3,350), (V4,300) }}	V1, V2, V3, V4, V7	V2-V1,V3-V1,V4-V1, V7-V2
4	{{(V7,650), (V3,350) },(V5,1600)}	V1, V2, V3, V4, V7, V5	V2-V1,V3-V1,V4-V1,V7-V2, V5-V4
5	{{(V7,650), (V6,450) },(V8,1250),(V5,1600)}	V1, V2, V3, V4, V7,V5, V6,V8	V2-V1,V3-V1,V4-V1,V7-V2, V6-V3,V8-V3
6	{{ (V7,650) },(V12,1150),(V8,1250),(V5,1600)}	V1, V2, V3, V4, V7,V5,V6,V8, V12	V2-V1,V3-V1,V4-V1,V7-V2,V6-V3,V8-V3, V12-V6
7	{{ (V12,1150) },(V8,1250),(V5,1600)}	V1, V2, V3, V4, V7,V5,V6,V8,V12	V2-V1,V3-V1,V4-V1,V7-V2,V6-V3,V8-V3,V12-V6
8	{{(V13,2100), (V8,1250) },(V5,1600)}	V1, V2, V3, V4, V7,V5,V6,V8,V12, V13	V2-V1,V3-V1,V4-V1,V7-V2,V6-V3,V8-V3,V12-V6, V13-V12

9	{(V13,2100),(V9,2040),(V10,1550),(V5,1600)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8
10	{(V13,2100),(V9,2040),(V5,1600)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8
11	{(V13,2100),(V9,2040),(V14,3000)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V5
12	{(V13,2100),(V11,3240),(V14,3000)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V5, V11-V9
13	{(V11,3240),(V14,2700)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V13 , V11-V9
14	{(V11,3240)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V13, V11-V9
15	{(V15,3640)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11, V15	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V13, V11-V9, V15-V11
16	{(V16,4940)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11, V15, V16	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V13, V11-V9, V15-V11, V16-V15
17	{(V17,5710)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11, V15, V16, V17	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V13, V11-V9, V15-V11, V16-V15, V17-V16
18	{(V18,6910)}	V1, V2, V3, V4, V7, V5, V6, V8, V12, V13, V9, V10, V14, V11, V15, V16, V17, V18	V2-V1, V3-V1, V4-V1, V7-V2, V6-V3, V8-V3, V12-V6, V13-V12, V9-V8, V10-V8, V14-V13, V11-V9, V15-V11, V16-V15, V17-V16, V18-V17

Đường đi ngắn nhất với UCS : V1-V3-V8-V9-V11-V15-V16-V17-V18. Tổng cost : 6910

Search Algorithm	DFS	BFS	UCS
Time complexity	$O(b^m)$	$O(b^d)$	$O(b^{(1+\lceil C^*/\epsilon \rceil)})$
Space complexity	$O(bm)$	$O(b^d)$	$O(b^{(1+\lceil C^*/\epsilon \rceil)})$

b : số node mà mỗi node expand được (branching factor)

m : độ sâu lớn nhất (maximum depth)

d: độ sâu của node mục tiêu tốt nhất (depth of shallowest solution)

C* : chi phí của giải pháp tối ưu (cost of optimal solution)

Epsilon : chi phí thấp nhất của từng hành động (every actions costs at least epsilon)

Nhận xét chung :

- Trường hợp tất cả các step cost đều bằng nhau thì BFS sẽ tối ưu vì BFS luôn expand giải pháp tốt nhất (shallowest solution) trước.
- DFS không tối ưu vì nó không cho giải pháp tốt nhất (shallowest solution) nếu nó tìm được một giải pháp không tối ưu (ex : deepest solution) trước .
- Về mặt memory thì DFS tốt hơn BFS.
- Trong đa số các trường hợp , UCS đều cho giải pháp tối ưu . Tuy nhiên , do chỉ expand dựa trên cost thấp nhất mà không có information về goal nên UCS có thể expand về mọi phía , do đó có thể dẫn đến rất tốn kém chi phí (expensive).
- Trường hợp tất cả step cost đều bằng nhau, UCS cũng trở thành BFS ngoại trừ việc BFS sẽ dừng khi tìm được goal ở độ sâu d trong khi UCS phải expand tất cả node ở độ sâu d để tìm goal có cost thấp nhất (việc này là không cần thiết), do đó BFS sẽ tốt hơn trong trường hợp này.

Nhận xét bài thực hành :

- Các thuật toán BFS, DFS, UCS khá là tương đồng , sự khác biệt ở đây là cách chúng ta expand child node (BFS -queue, DFS-stack , UCS -priority queue)
- Ở trong đoạn code hàm UCS cần bổ sung trường hợp node đã visited nhưng có cost lớn hơn cost chúng ta đang xét thì cập nhật lại cost và parent của node đó.

```

if node not in visited:
    frontier.put((current_weight+weight, node))
    parent[node] = current_node
    visited.append(node)
else:
    #nếu node đã nằm trong visited mà có old_weight > current_weight+weight thì
    # cập nhập lại frontier và parent
    for item in frontier.queue:
        if node == item[1] and current_weight+weight < item[0]:
            frontier.queue.remove(item)
            frontier.put((current_weight+weight, node))
            parent[node] = current_node
            break

```

- Trong bài này DFS cho kết quả tốt nhất (nhanh nhất), BFS và UCS cho kết quả như nhau.