



dfs.ipynb

File Edit Lihat Sisipkan Runtime Fitur Bantuan



Perintah + Kode + Teks



```
self.DFSutil(v, visited)

# Kode pengguna
if __name__ == "__main__":
    g = Graph()
    g.addEdge(0, 1)
    g.addEdge(0, 2)
    g.addEdge(1, 2)
    g.addEdge(2, 0)
    g.addEdge(2, 3)
    g.addEdge(3, 3)

    print("Berikut adalah Penelusuran Depth First (dimulai dari node 2):")
    g.DFS(2)
```

Berikut adalah Penelusuran Depth First (dimulai dari node 2):
2 0 1 3



bfs.ipynb

File Edit Lihat Sisipkan Runtime Fitur Bantuan



Perintah + Kode + Teks



```
while queue:
    # Mengambil elemen dari antrian
    vertex = queue.popleft()
    print(vertex, end=" ") # Mencetak node yang dikunjungi

    # Memproses semua tetangga yang belum dikunjungi
    for neighbour in graph[vertex]:
        if neighbour not in visited:
            visited.add(neighbour)
            queue.append(neighbour)

# Kode utama
if __name__ == "__main__":
    graph = {0: [1, 2], 1: [2], 2: [3], 3: [1, 2]}
    print("Following is Breadth First Traversal:")
    bfs(graph, 0)
```

Following is Breadth First Traversal:
0 1 2 3

Salinan dari Sandra...

ucs.ipynb ☆

File Edit Lihat Sisipkan Runtime Fitur Bantuan

Q Perintah + Kode + Teks

```
goal = []
goal.append(6)

# Get the answer
answer = uniform_cost_search(goal, 0)

# Print the minimum cost from node 0 to node 6
print("Minimum cost from 0 to 6 is =", answer[0])
```

Minimum cost from 0 to 6 is = 3

greedy_best_first.ipynb ☆

File Edit Lihat Sisipkan Runtime Fitur Bantuan

Q Perintah + Kode + Teks

```
'B': {'D'},
'C': {'G'},
'D': {'G'},
'E': {'D'}
}

# Titik awal dan tujuan
start_node = 'S'
goal_node = 'G'

# Panggil fungsi greedy search
greedy_search(graph, start_node, goal_node)
```

Simpul tujuan sudah ditemukan!
True



Salinan dari Sandra...



A star tree.ipynb ☆ ☁

File Edit Lihat Sisipkan Runtime Fitur Bantuan



Q Perintah + Kode + Teks



{x}



```
# Graf (dalam bentuk daftar kejadian dengan bobot)
graph = {
    'S': {'A': 3, 'E': 2},
    'A': {'B': 3, 'C': 4},
    'B': {'D': 5},
    'C': {'G': 7},
    'D': {'G': 3},
    'E': {'D': 6}
}

# Titik awal dan tujuan
start_node = 'S'
goal_node = 'G'

# Panggil fungsi A* search
a_star_search(graph, start_node, goal_node, heuristic)
```



Simpul tujuan sudah ditemukan!
True

<>



Salinan dari Sandra...



A star Graph.ipynb ☆ ☁

File Edit Lihat Sisipkan Runtime Fitur Bantuan



Q Perintah + Kode + Teks



{x}



```
# Graf (dalam bentuk daftar kejadian dengan bobot)
graph = {
    'S': {'A': 3, 'E': 2},
    'A': {'B': 3, 'C': 4},
    'B': {'D': 5},
    'C': {'G': 7},
    'D': {'G': 3},
    'E': {'D': 6}
}

# Titik awal dan tujuan
start_node = 'S'
goal_node = 'G'

# Panggil fungsi A* search
a_star_search(graph, start_node, goal_node, heuristic)
```



Simpul tujuan sudah ditemukan!
True

<>

