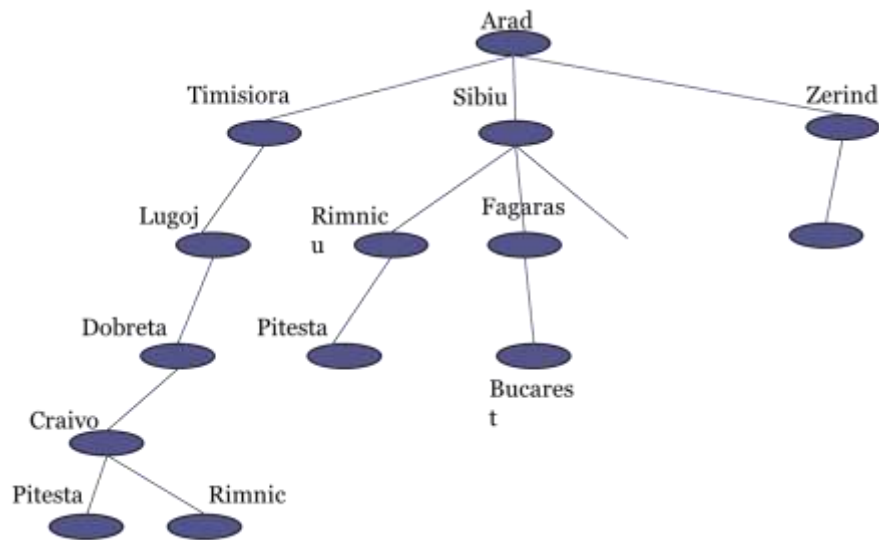


EVALUASI

Case 1:



- Bagaimana rute perjalanan dari *Arad* ke *Bucarest*
- Gunakan teknik pencarian *Breadth – First Search* & *Depth – First Search*
- Lakukan penyederhanaan *Tree* dengan pohon OR dan pohon AND

Case 2:

Sebuah puzzle berukuran 3X3

Nilai awal :

1	2	3
8		4
7	6	5

Goal :

2	8	3
1	6	4
7		5

$f(n) = g(n) + h(n)$ $g(n)$ = kedalaman

pohon $h(n)$ = jumlah angka yang

salah posisi

Kerjakan dengan Teknik Best First Search

1.

a. Arad \rightarrow Sibiu \rightarrow Fagaras \rightarrow Bucurest.

b. BFS:

1. Mulai dari node Arad.
2. Tambahkan semua tetangga Arad (Sibiu, Timisoara, Zerind) ke antrian.
3. Ambil node pertama dari antrian (misalnya Sibiu).
4. Tambahkan semua tetangga Sibiu yang belum dikunjungi ke antrian (Fagaras, Rimnicu Vilcea).
5. Ulangi langkah 3 dan 4 sampai Bucharest ditemukan atau antrian kosong.

DFS:

1. Mulai dari node Arad.
2. Pilih salah satu tetangga Arad (misalnya Sibiu) dan lanjutkan ke node tersebut.
3. Jika node yang dipilih adalah Bucharest, maka pencarian selesai.
4. Jika tidak, pilih salah satu tetangga dari node saat ini dan lanjutkan pencarian secara rekursif.
5. Jika semua cabang dari suatu node sudah dieksplorasi, kembali ke node sebelumnya dan coba cabang lain.

c. Pohon OR

- Setiap percabangan dalam grafik dapat diwakili oleh sebuah node OR.
- Misalnya, dari node Arad, ada tiga pilihan: ke Sibiu, Timisoara, atau Zerind. Ini dapat direpresentasikan sebagai sebuah node OR dengan tiga anak.

Pohon AND

- Jika ada kondisi yang harus dipenuhi secara bersamaan untuk mencapai tujuan, maka dapat direpresentasikan dengan node AND.
- Misalnya, untuk mencapai Bucharest dari Arad, kita harus melewati beberapa kota secara berurutan. Ini dapat direpresentasikan sebagai serangkaian node AND.

AWAL

1	2	3
8		4
7	6	5

GOAL

2	8	3
1	6	4
7		5

Ubin yang salah posisi: 1, 2, 8, 4, 6 (total 5 salah posisi).

$$G = 0, H = 5, F = G + H = 0 + 5 = 5$$

Langkah 1: Geser kosong ke kiri

Konfigurasi baru:

markdown

1	2	3
	8	4
7	6	5

Posisi kosong: (2,1).

Ubin yang salah posisi: 1, 2, 8, 4, 6 (total 5 salah posisi).

$$G = 1, H = 5, F = G + H = 1 + 5 = 6$$

Langkah 2: Geser kosong ke atas

Konfigurasi baru:

	2	3
1	8	4
7	6	5

Posisi kosong: (1,1).

Ubin yang salah posisi: 2, 8, 4, 6 (total 4 salah posisi).

$$G = 2, H = 4, F = G + H = 2 + 4 = 6$$

Langkah 3: Geser kosong ke kanan

Konfigurasi baru:

	2	3
1	8	4
7	6	5

Posisi kosong: (1,2).

Ubin yang salah posisi: 8, 4, 6 (total 3 salah posisi).

$$G = 3, H = 3, F = G + H = 3 + 3 = 6$$

Langkah 4: Geser kosong ke bawah

Konfigurasi baru:

2	8	3
1		4
7	6	5

Posisi kosong: (2,2).

Ubin yang salah posisi: 4, 6 (total 2 salah posisi).

$$G = 4, H = 2, F = G + H = 4 + 2 = 6$$

Langkah 5: Geser kosong ke kanan

Konfigurasi baru:

2	8	3
1	6	
7		5

Posisi kosong: (2,3).

Ubin yang salah posisi: 6, 5 (total 2 salah posisi).

$$G = 5, H = 2, F = G + H = 5 + 2 = 7$$

Langkah 6: Geser kosong ke bawah

Konfigurasi baru:

2	8	3
1	6	4
7		5

Posisi kosong: (3,3).

Ubin yang salah posisi: Semua di tempatnya.

$$G = 6, H = 0, F = G + H = 6 + 0 = 6$$

Goal state tercapai!

2	8	3
1	6	4
7		5

Kesimpulan:

Total langkah yang diambil: 6 langkah. Teknik Best First

Search memastikan $f(n)$ minimum setiap kali.