

Penerapan BFS dan DFS pada Pencarian Solusi

Bahan Kuliah IF2151 Strategi Algoritmik

Oleh: Rinaldi Munir

Pengorganisasian Solusi

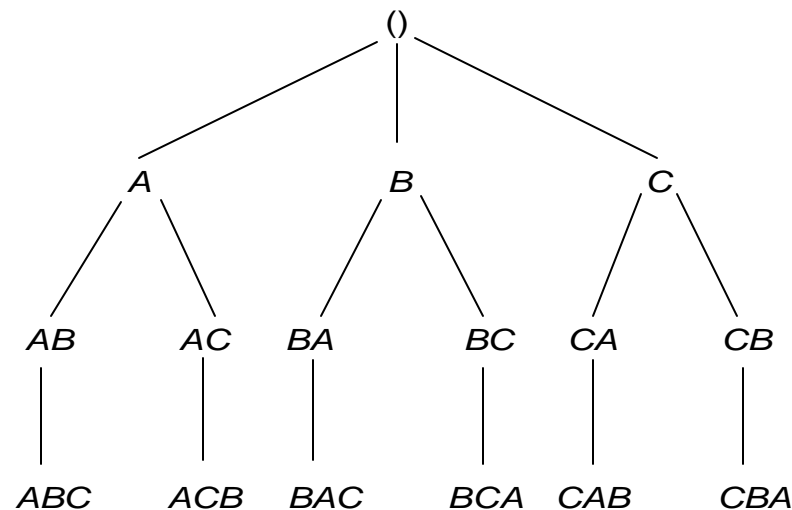
- Kemungkinan2 solusi dari persoalan membentuk ruang solusi (*solution space*)
- Ruang solusi diorganisasikan ke dalam struktur pohon
- Pencarian solusi dilakukan dengan mengunjungi (traversal) simpul-simpul di dalam pohon

- Pohon yang ditelusuri: pohon dinamis
- Pohon dinamis: pohon yang dibangun selama pencarian solusi berlangsung
- Pohon dinamis dibangun dengan 2 metode traversal: BFS dan DFS
- Pohon dinamis menyatakan status-status persoalan pada saat pencarian solusi berlangsung.

Terminologi

- **Status persoalan** (*problem state*): simpul-simpul di dalam pohon dinamis yang memenuhi kendala (*constraints*).
- **Status solusi** (*solution state*): satu atau lebih status yang menyatakan solusi persoalan.
- **Status tujuan** (*goal state*): status solusi yang merupakan simpul daun
- **Ruang solusi** (*solution space*): himpunan semua status solusi.
- **Ruang status** (*state space*): Seluruh simpul di dalam pohon dinamis dan pohonnya dinamakan juga **pohon ruang status** (*state space tree*).

Contoh 6.1. Pohon ruang status yang dibangun untuk menghasilkan semua permutasi A, B, C :



Ket: $()$ = status kosong

- Setiap simpul di dalam pohon menyatakan status persoalan.
- Status awal adalah akar yang berupa sebuah “kosong”.
- Setiap daun pada pohon tersebut (ABC, ACB, BAC, BCA, CAB, dan CBA) menyatakan status solusi, dan semua daun adalah ruang solusi.

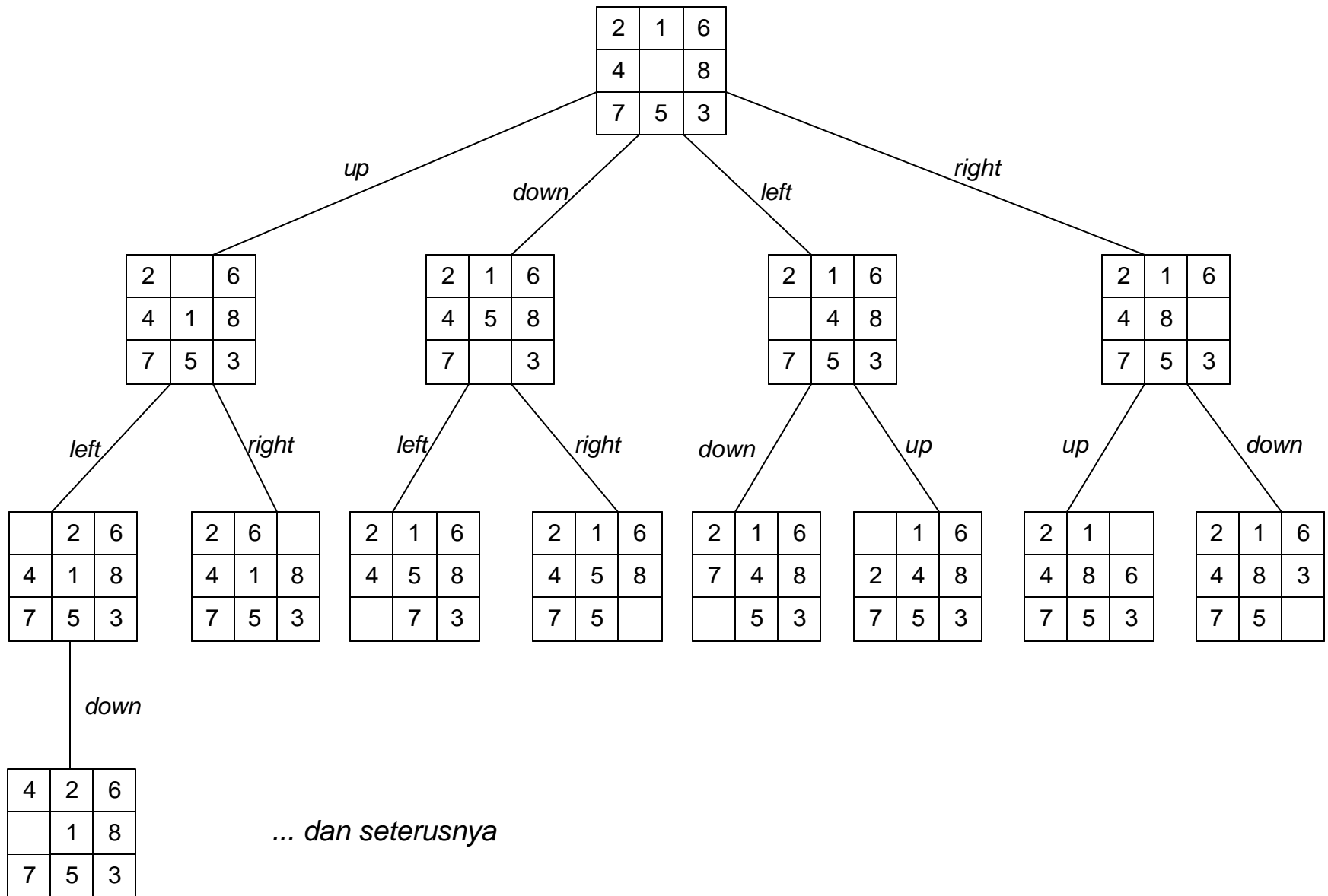
Contoh 6.2. Permainan 8-puzzle:

2	1	6
4		8
7	5	3

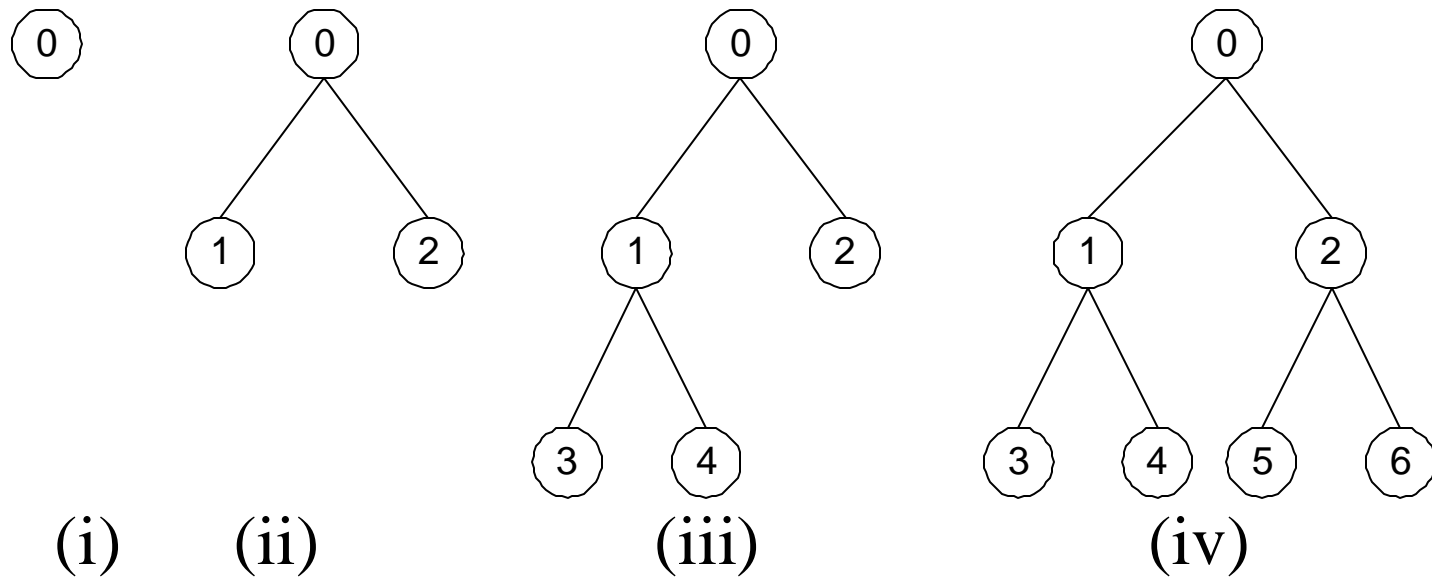
(a) Susunan awal
(*initial state*)

1	2	3
8		4
7	6	5

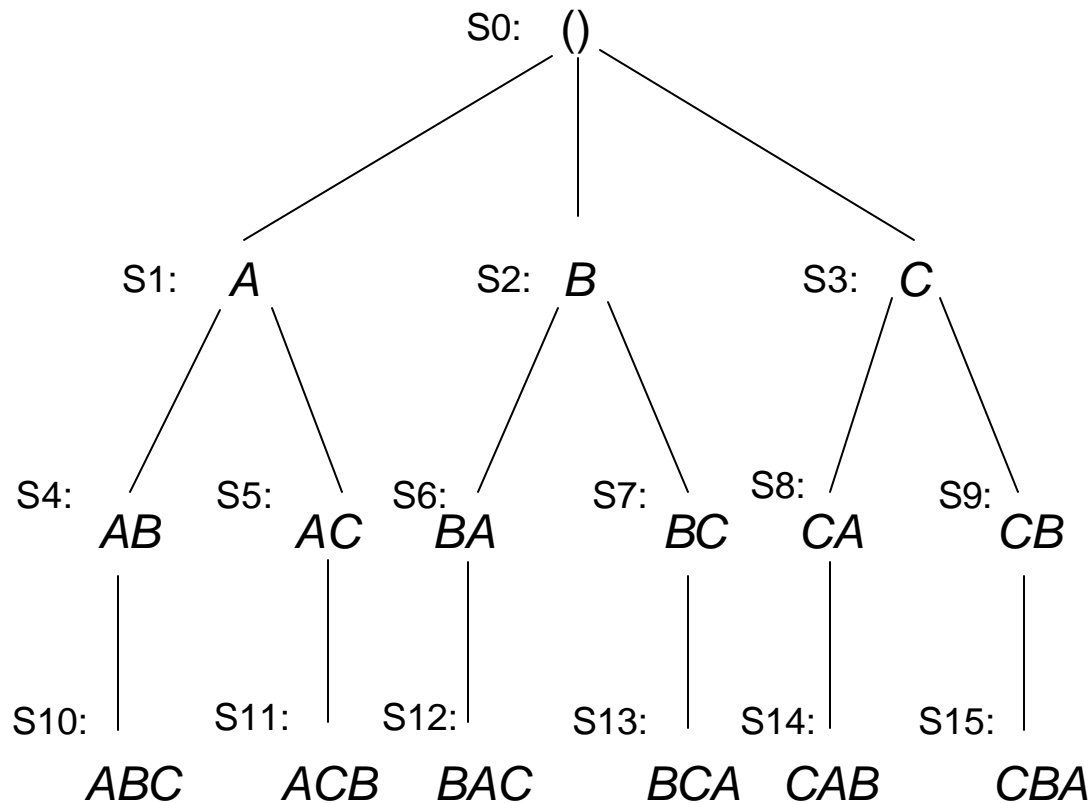
(b) Susunan akhir
(*goal state*)



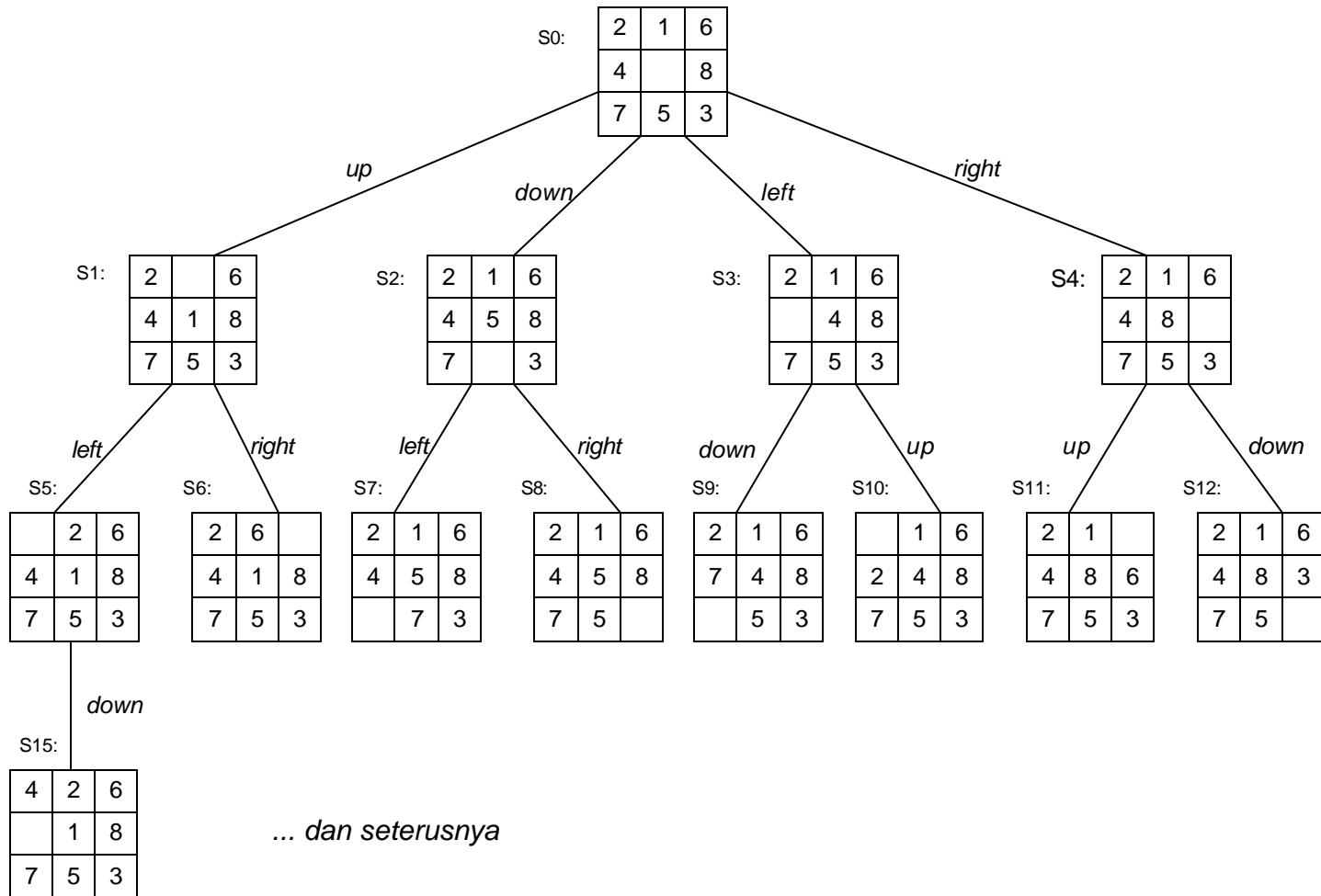
Metode Pencarian Melebar (BFS)



Gambar 6.4. Tahapan pembentukan pohon BFS

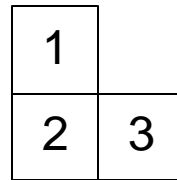


Gambar 6.5 Pembentukan pohon ruang status persoalan pembangkitan permutasi A, B, C dengan metode BFS

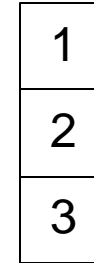


Gambar 6.6 Pembentukan pohon ruang status persoalan 8-puzzle dengan metode BFS.

Contoh 6.3. Sebuah mainan yang terdiri atas 3 buah blok (dinomori 1, 2, dan 3).



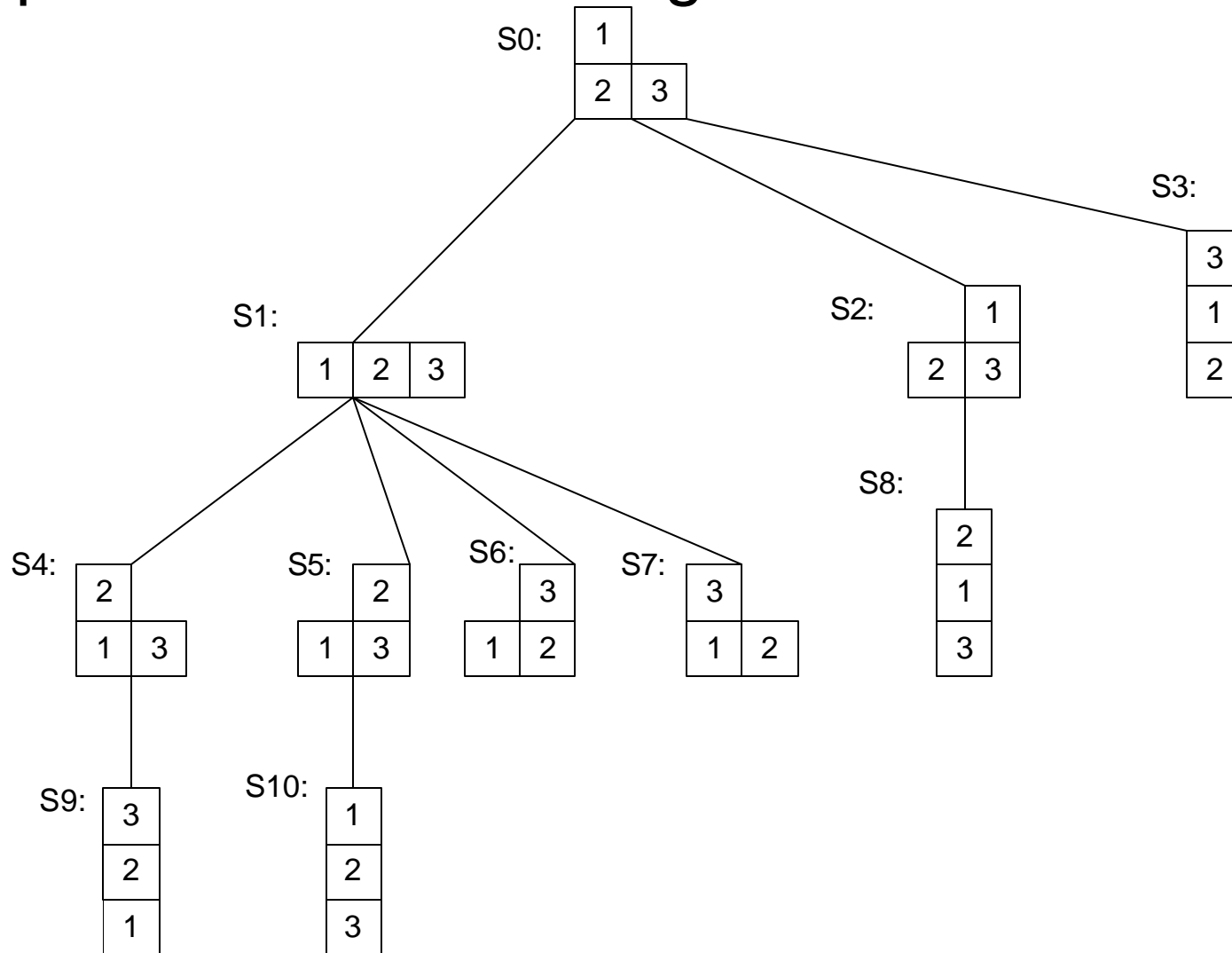
(a) Susunan awal



(b) Susunan akhir

- Operator perpindahan: “PINDAHKAN X ke Y”, yang berarti memindahkan objek X ke atas objek yang lain.
- Pada setiap saat, hanya satu buah blok yang boleh dipindahkan.
- Operator tidak digunakan untuk membangkitkan status yang sama lebih dari satu kali.

Pohon ruang status yang dibentuk selama pencarian solusi dengan metode BFS:

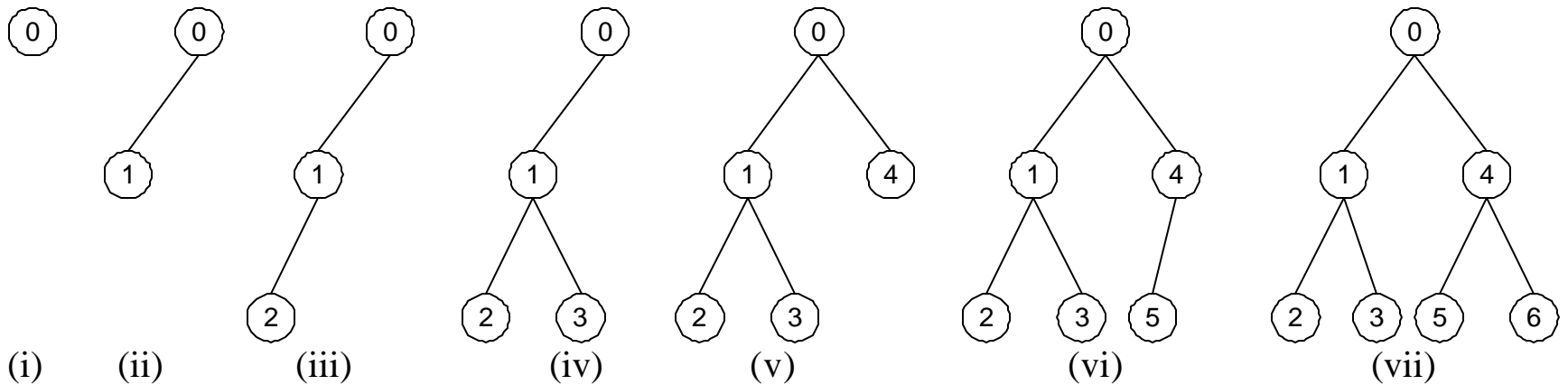


- Dengan mengikuti lintasan dari simpul akar (S_0) ke simpul solusi (S_{10}), kita memperoleh konfigurasi urutan perpindahan blok dari status awal sampai ke status akhir.
- Dengan metode BFS, jika terdapat sebuah solusi, maka BFS menjamin dapat menemukannya.
- Jika terdapat lebih dari satu buah solusi, BFS selalu menemukan solusi pertama pada aras pohon yang paling rendah.

Kompleksitas waktu algoritma BFS:

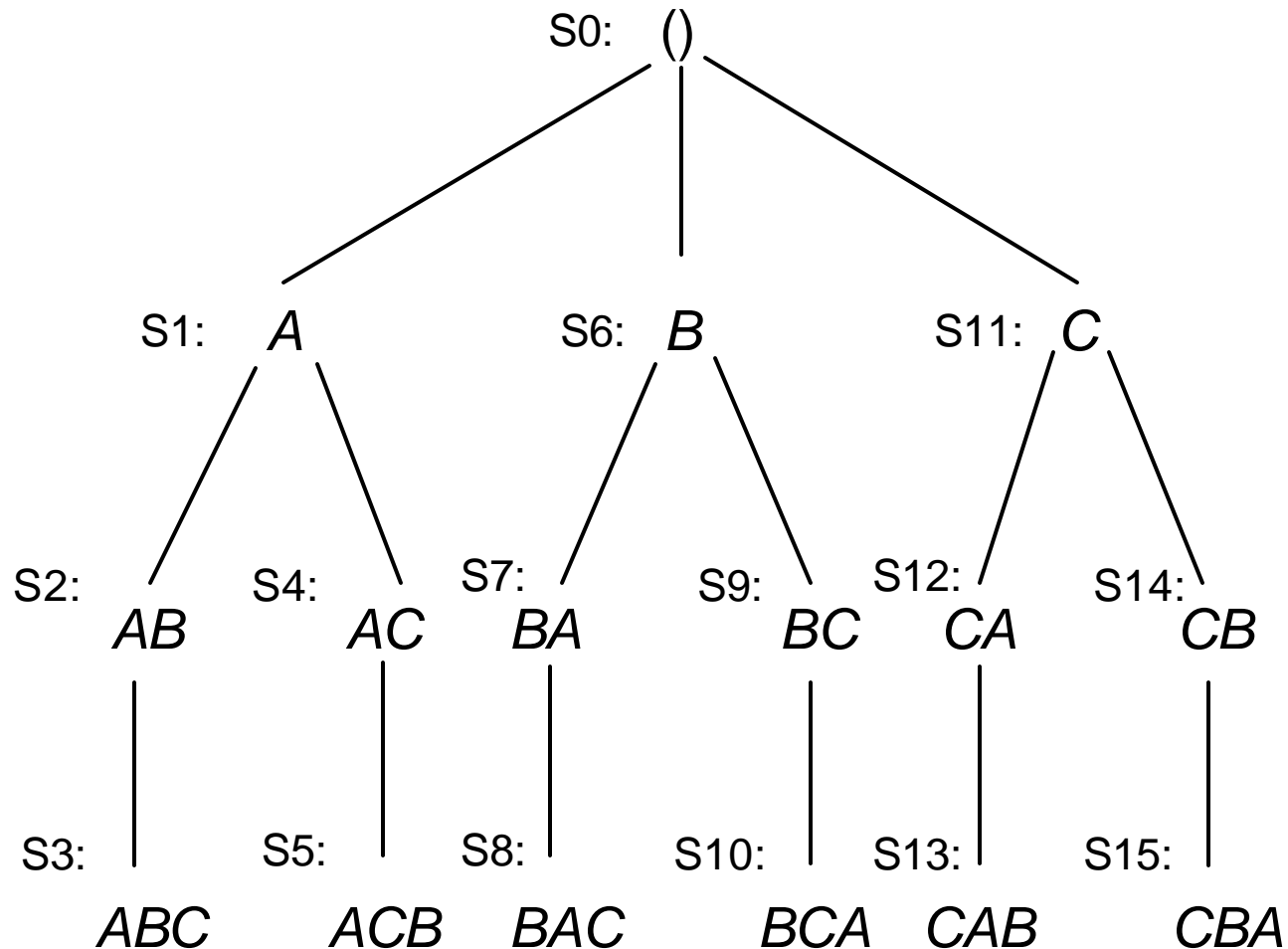
- Asumsi: setiap simpul dapat membangkitkan b buah simpul baru.
- Misalkan solusi ditemukan pada aras ke- d
- Jumlah maksimum seluruh simpul:
$$1 + b + b^2 + b^3 + \dots + b^d = (b^{d+1} - 1)/(b - 1)$$
- $T(n) = O(b^d)$.
- Kompleksitas ruang algoritma BFS = sama dengan kompleksitas waktunya, karena semua simpul daun dari pohon harus disimpan di dalam memori selama proses pencarian.

Metode Pencarian Mendalam (DFS)



Gambar 6.9. Tahapan pembentukan pohon DFS

Pembentukan pohon ruang status persoalan pembangkitan permutasi A, B, C dengan metode DFS



- **Contoh.** Sebuah bidak (pion) bergerak di dalam sebuah matriks pada Gambar 6.11. Bidak dapat memasuki elemen matriks mana saja pada baris paling atas. Dari elemen matriks yang berisi 0, bidak dapat bergerak ke bawah jika elemen matriks di bawahnya berisi 0; atau berpindah horizontal (kiri atau kanan) jika elemen di bawahnya berisi 1. Bila bidak berada pada elemen yang berisi 1, ia tidak dapat bergerak kemanapun. Tujuan permainan ini adalah mencapai elemen matriks yang mengandung 0 pada baris paling bawah.

	1	2	3	4
1	1	0	0	0
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0

Gambar 6.11 Matriks bidak

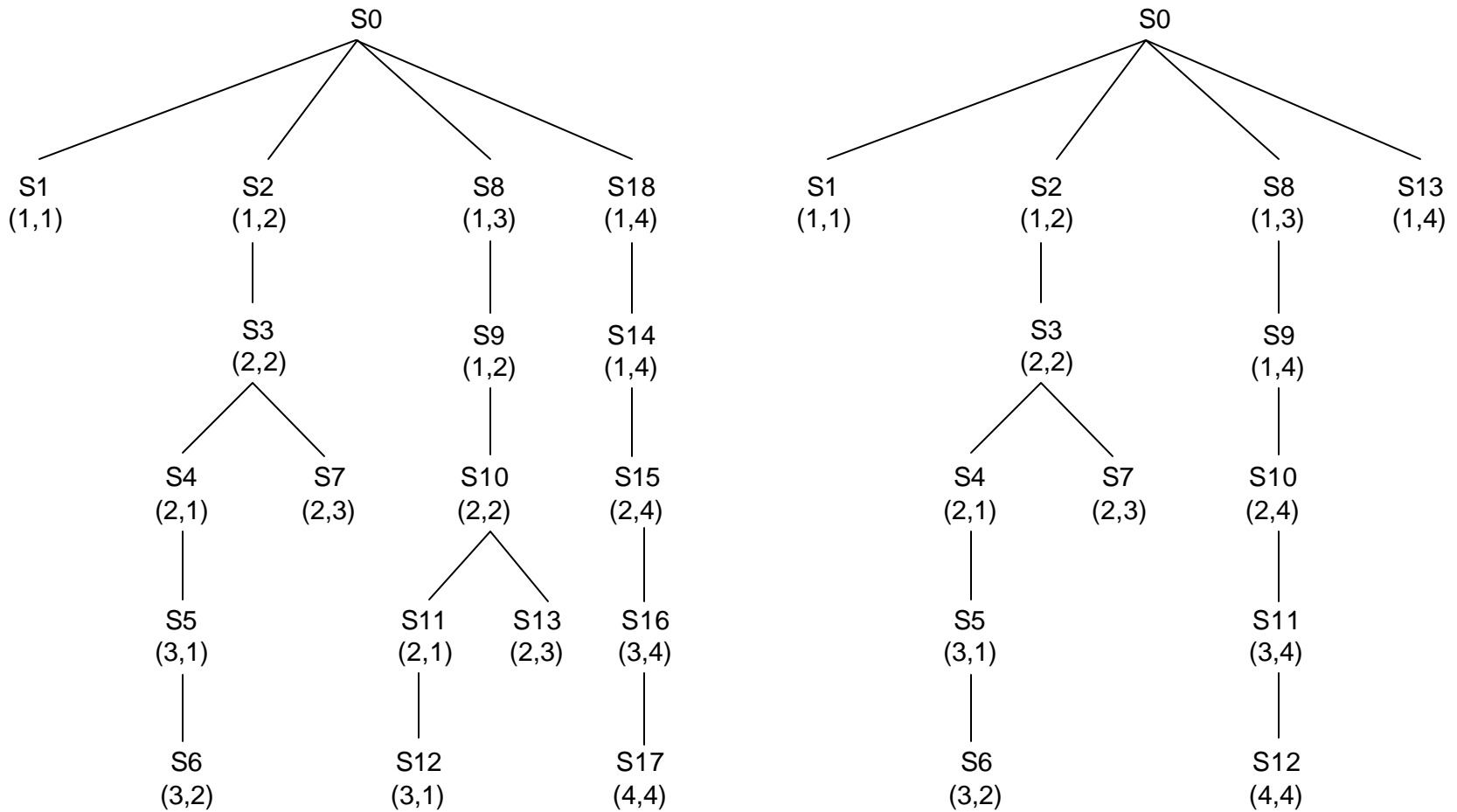
Operator yang digunakan:

DOWN pindahkan bidak satu posisi ke bawah

LEFT pindahkan bidak satu posisi ke kiri

RIGHT pindahkan bidak satu posisi ke kanan

Batas kedalaman maksimum pohon ruang status diandaikan 5.



Gambar 6.12

- (a) Pohon ruang status yang mengandung duplikasi simpul
 (b) Pohon ruang status yang menghindari pembangkitan simpul yang sama.

- Kompleksitas waktu algoritma DFS pada kasus terburuk adalah $O(bm)$.
- Kompleksitas ruang algoritma DFS adalah $O(bm)$, karena kita hanya perlu menyimpan satu buah lintasan tunggal dari akar sampai daun, ditambah dengan simpul-simpul saudara kandungnya yang belum dikembangkan.

- Untuk persoalan yang memiliki banyak solusi, metode DFS lebih cepat daripada BFS, karena DFS menemukan solusi setelah mengeksplorasi hanya sebagian kecil dari seluruh ruang status.
- Sedangkan BFS masih harus menelusuri semua lintasan pada aras $d - 1$ sebelum memeriksa solusi pada aras d .

Varian DFS: Metode Pencarian Mendalam Berulang (IDS = *Iterative Deepening Search*)

- Kesulitan utama pada metode DFS adalah menentukan batas maksimum kedalaman pohon ruang status.
- Strategi yang digunakan untuk memecahkan masalah kedalaman terbaik ini adalah dengan mencoba semua kedalaman yang mungkin, mula-mula kedalaman 0, kedalaman 1, kedalaman 2, dan seterusnya.