



Algoritma *Branch and Bound*

(Bagian 1)



Algoritma *Branch and Bound*

- Algoritma *Branch and Bound* (B&B) juga merupakan metode pencarian di dalam ruang solusi secara sistematis.
- Algoritma runut-balik → skema DFS
- Algoritma B&B → skema BFS

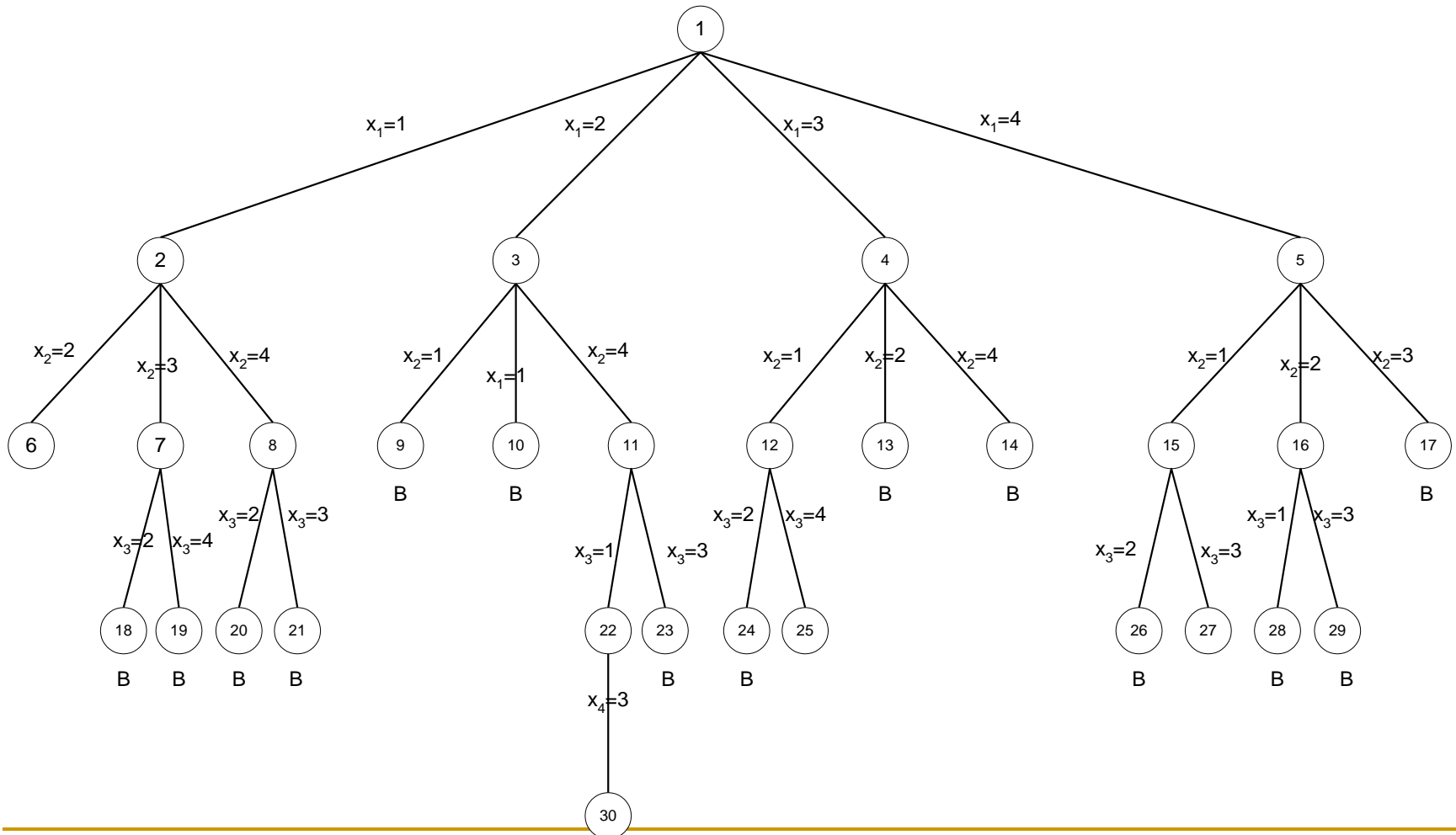
- Untuk mempercepat pencarian ke simpul solusi, maka setiap simpul diberi sebuah nilai ongkos (*cost*).
- Simpul berikutnya yang akan diekspansi tidak lagi berdasarkan urutan pembangkitannya (sebagaimana pada BFS murni), tetapi simpul yang memiliki ongkos yang paling kecil (*least cost search*).

-
- Nilai ongkos pada setiap simpul i menyatakan taksiran ongkos termurah lintasan dari simpul i ke simpul solusi (*goal node*):

$\hat{c}(i)$ = nilai taksiran lintasan termurah dari
simpul status i ke status tujuan

- Dengan kata lain, $\hat{c}(i)$ menyatakan batas bawah (*lower bound*) dari ongkos pencarian solusi dari status i .
-

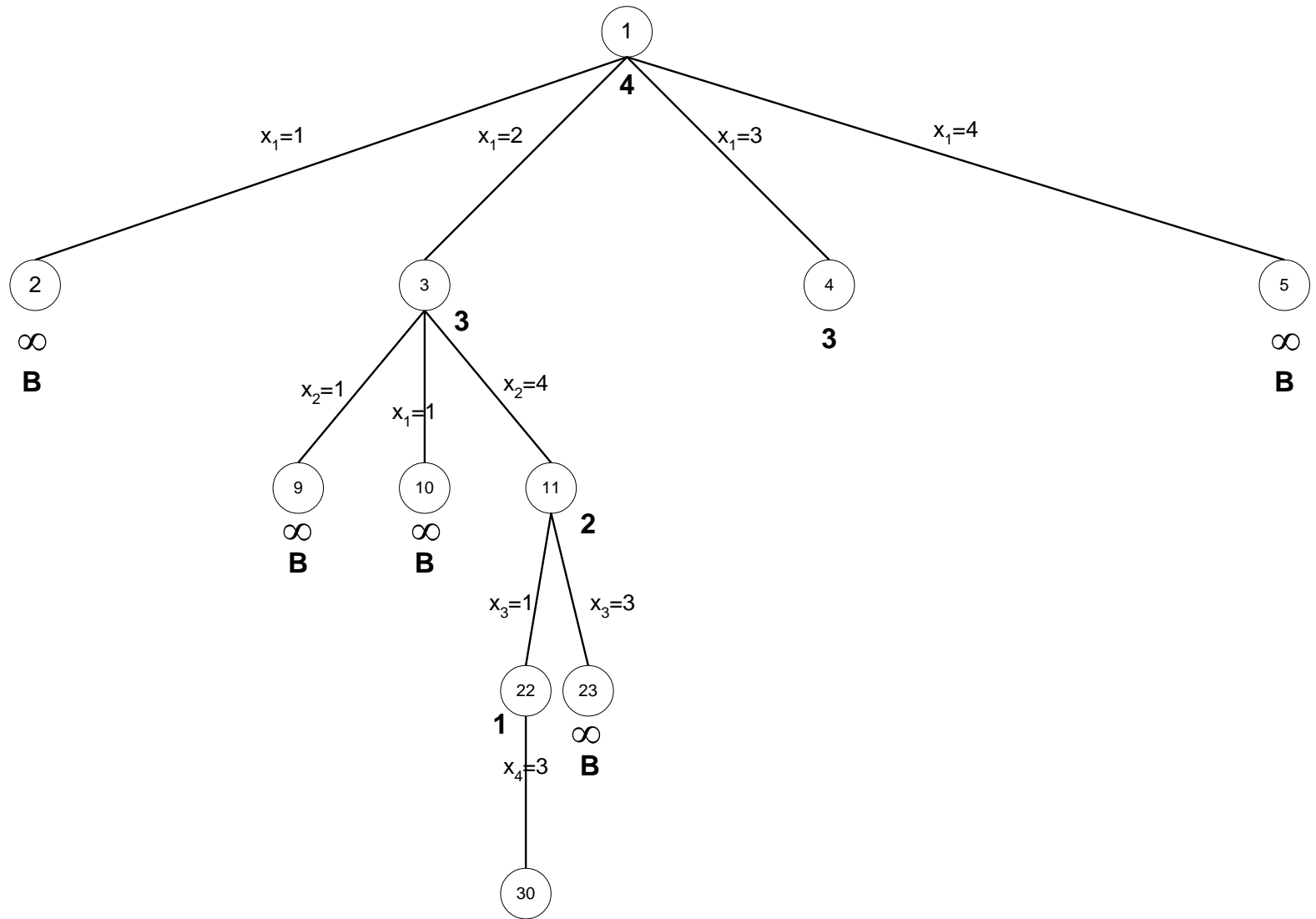
Tinjau kembali persoalan 4-ratu yang diselesaikan dengan skema BFS (murni).



- Solusi pertama dicapai pada simpul 30, yaitu $X = (2, 4, 1, 3)$. Dengan skema BFS murni / FIFO, kita harus memperluas dulu simpul 12, simpul 15, dan simpul 16 sebelum memperluas simpul 22 yang melahirkan simpul solusi, yaitu simpul 30.
- Pada algoritma B&B, pencarian ke simpul solusi dapat dipercepat dengan memilih simpul hidup berdasarkan nilai ongkos (*cost*).

-
- Setiap simpul hidup diasosiasikan dengan sebuah ongkos yang menyatakan nilai batas (*bound*).
 - Simpul hidup yang menjadi simpul-E ialah simpul yang mempunyai nilai batas terkecil (strategi **pencarian berdasarkan biaya terkecil** (*least cost search*)).
-

-
- Untuk setiap simpul X , nilai batas ini dapat berupa [HOR78]:
 1. jumlah simpul dalam upapohon X yang perlu dibangkitkan sebelum simpul solusi ditemukan, atau
 2. panjang lintasan dari simpul X ke simpul solusi terdekat (dalam upapohon X ybs)
 - Misal digunakan ukuran (b):
-



-
- Pemberian nilai batas seperti pada persoalan N-Ratu di atas adalah nilai batas yang ideal, karena letak simpul solusi diketahui.
 - Pada umumnya, untuk kebanyakan persoalan, letak simpul solusi tidak diketahui, karena itu, dalam prakteknya, nilai batas untuk setiap simpul umumnya berupa taksiran atau perkiraan.
-

- Fungsi heuristik untuk menghitung taksiran *cost*:

$$\hat{c}(i) = \hat{f}(i) + \hat{g}(i)$$

$\hat{c}(i)$ = ongkos untuk simpul i

$\hat{f}(i)$ = ongkos mencapai simpul i dari akar

$\hat{g}(i)$ = ongkos mencapai simpul tujuan dari simpul i .

- Simpul berikutnya yang dipilih untuk diekspansi adalah simpul yang memiliki \hat{c} minimum.

Algoritma B&B:

1. Masukkan simpul akar ke dalam antrian Q . Jika simpul akar adalah simpul solusi (*goal node*), maka solusi telah ditemukan. Stop.
2. Jika Q kosong, tidak ada solusi. Stop.
3. Jika Q tidak kosong, pilih dari antrian Q simpul i yang mempunyai $\hat{c}(i)$ paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih satu secara sembarang.
4. Jika simpul i adalah simpul solusi, berarti solusi sudah ditemukan, stop. Jika simpul i bukan simpul solusi, maka bangkitkan semua anak-anaknya. Jika i tidak mempunyai anak, kembali ke langkah 2.
5. Untuk setiap anak j dari simpul i , hitung $\hat{c}(j)$, dan masukkan semua anak-anak tersebut ke dalam Q .
6. Kembali ke langkah 2.

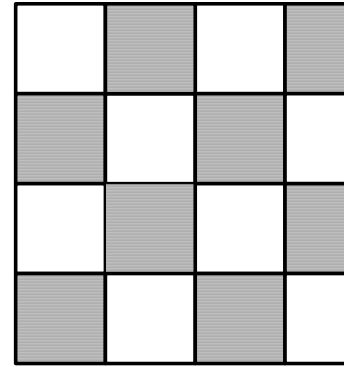
Permainan 15-*Puzzle*

1	3	4	15
2		5	12
7	6	11	14
8	9	10	13

(a) Susunan awal

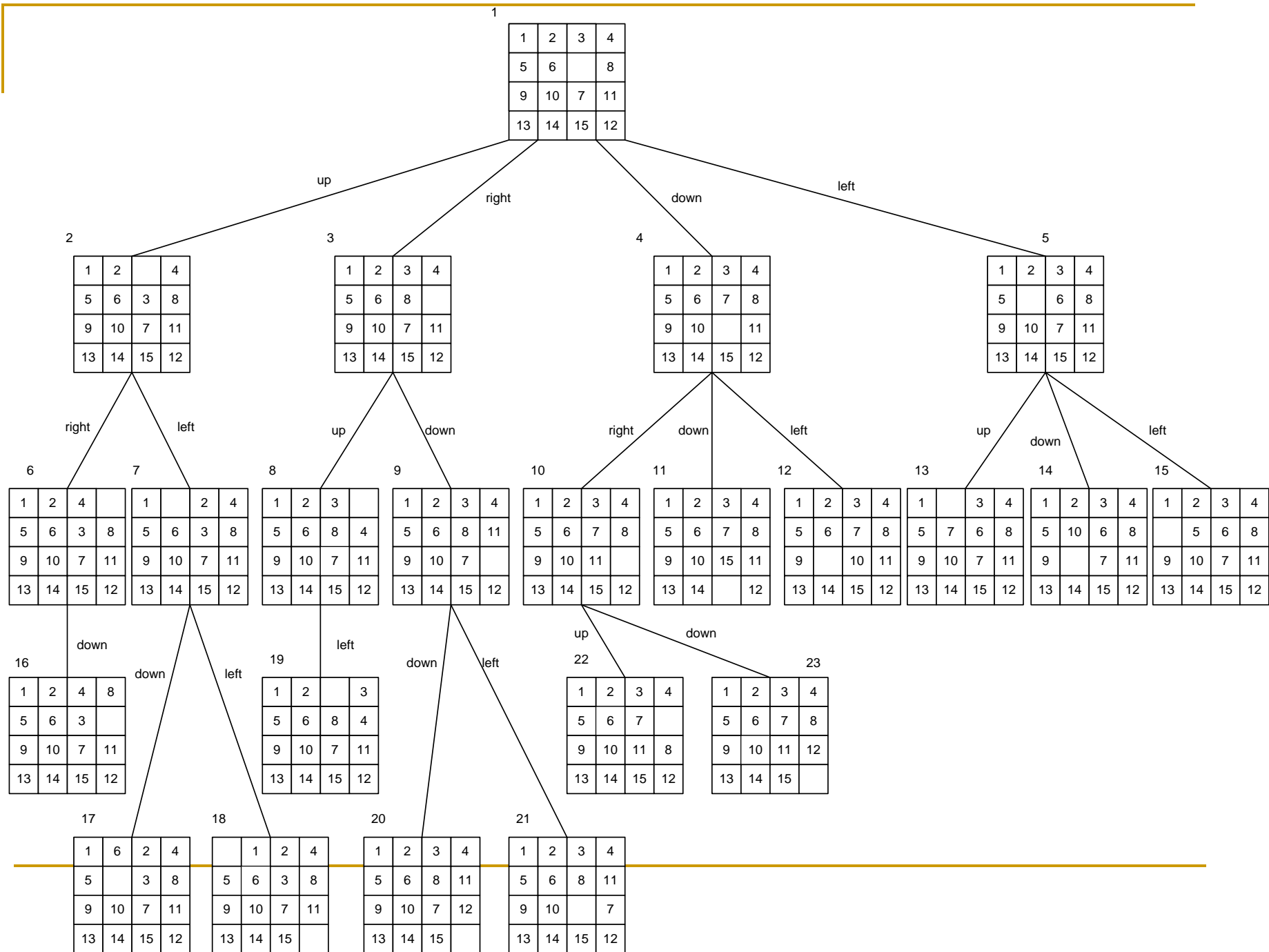
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

(b) Susunan akhir



(c)

- Terdapat $16!$ ($= 20,9 \times 10^{12}$) susunan ubin yang berbeda pada bidang kerangka



- Sebelum menelusuri ruang status untuk mencapai susunan akhir, kita patut menentukan apakah status tujuan dapat dicapai atau tidak dari status awal.
- $POSIS(i)$ = posisi ubin bernomor i pada susunan akhir.
- $KURANG(i)$ = jumlah ubin j sedemikian sehingga $j < i$ dan $POSIS(j) > POSIS(i)$.

1	3	4	15
2		5	12
7	6	11	14
8	9	10	13

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Misalkan $X = 1$ jika pada status awal slot kosong berada pada salah satu posisi yang diarsir pada Gambar 7.3c, dan $X = 0$ jika slot kosong berada pada posisi lainnya.

- **Teorema 8.1.** Status tujuan hanya dapat dicapai dari status awal jika

$$\sum_{i=1}^{16} KURANG(i) + X$$

bernilai genap.

- Pada Gambar 7.2a mempunyai $X = 0$ dan $\sum_{i=1}^{16} KURANG(i) = 37$, sehingga $37 + 0 = 37$ (ganjil).
- Oleh karena itu, status tujuan tidak dapat dicapai dari status awal pada Gambar 7.2a.

- **Algoritma B&B:**

Nilai ongkos untuk simpul P : $\hat{c}(P) = f(P) + \hat{g}(P)$

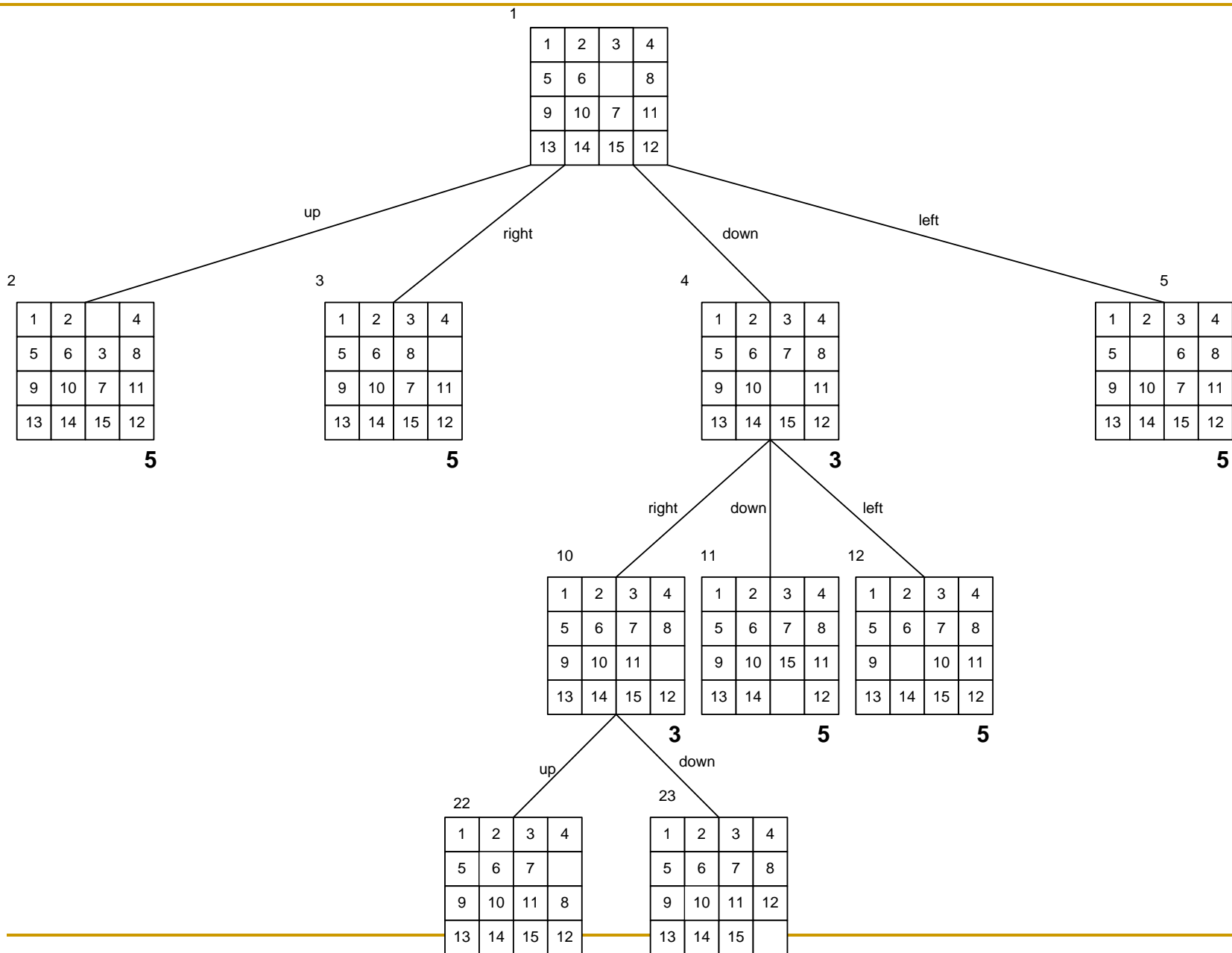
$f(P)$ = adalah panjang lintasan dari simpul akar ke P

$\hat{g}(P)$ = taksiran panjang lintasan terpendek dari P ke simpul solusi pada upapohon yang akarnya P .

-
- Salah satu cara menghitung $\hat{g}(P)$:

$\hat{g}(P)$ = jumlah ubin tidak kosong yang tidak terdapat pada susunan akhir

- Paling sedikit sejumlah $\hat{g}(P)$ perpindahan harus dilakukan untuk mentransformasikan status P ke status tujuan.
-



**simpul
solusi**

Berlanjut....
