



IN244 – Strategi Algoritmik

09 – Branch & Bound

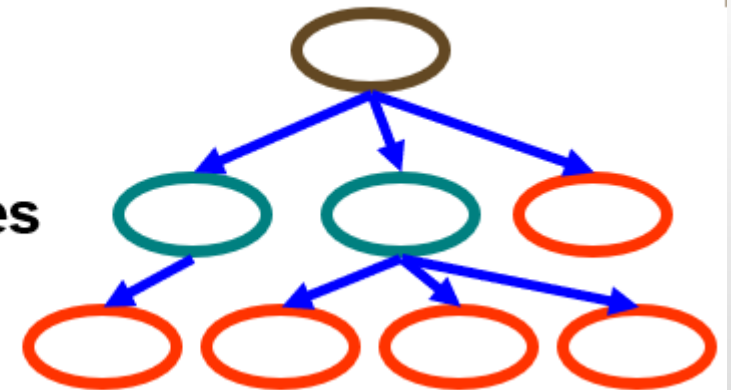
Struktur Data

- Algoritma branch & bound memakai struktur tree dalam pencarian solusi
- Tree : struktur graf yang tidak mengandung sirkuit
- Traversal yang dipakai dalam backtracking
 - Pencarian Melebar
 - Breadth First Search

Root node

Interior nodes

Leaf nodes



Algoritma Branch & Bound (B&B)

- Digunakan untuk persoalan optimisasi, yaitu meminimalkan atau memaksimalkan suatu fungsi objektif, yang tidak melanggar Batasan (*constraints*) persoalan
- Branch & Bound = BFS yang dilengkapi *least cost search*
- BFS murni: Simpul berikutnya yang akan diekspansi berdasarkan urutan pembangkitannya (FIFO)
- 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), tapi
 - Berdasarkan 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*), yaitu
 - nilai taksiran lintasan termurah dari simpul status i ke status tujuan
 - batas bawah (*lower bound*) dari ongkos pencarian solusi status i .

Prinsip Pencarian Solusi

- Algoritma B&B juga menerapkan “pemangkasan” pada jalur yang dianggap tidak lagi mengarah pada solusi.
- Kriteria pemangkasan secara umum:
 - Nilai simpul tidak lebih baik dari nilai terbaik sejauh ini (*the best solution so far*)
 - Simpul tidak merepresentasikan solusi yang ‘feasible’ karena ada batasan yang dilanggar
 - Solusi pada simpul tersebut hanya terdiri atas satu titik, sehingga tidak ada pilihan lain; bandingkan nilai fungsi obyektif dengan solusi terbaik saat ini, yang terbaik yang diambil

Prinsip pencarian solusi

- 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

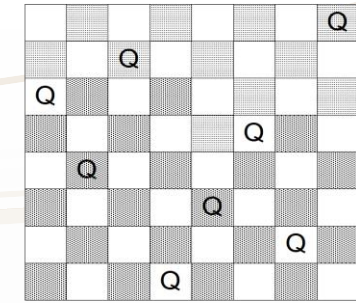
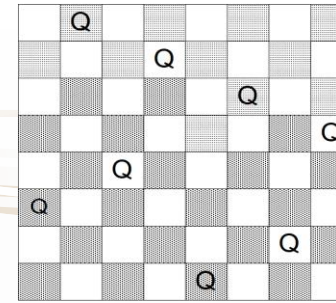
$\hat{g}(i)$ = ongkos mencapai simpul tujuan dari simpul i .

- Simpul berikutnya yang dipilih untuk diekspansi adalah simpul yang memiliki *cost* minimum.

Tahapan Algoritma Branch & Bound

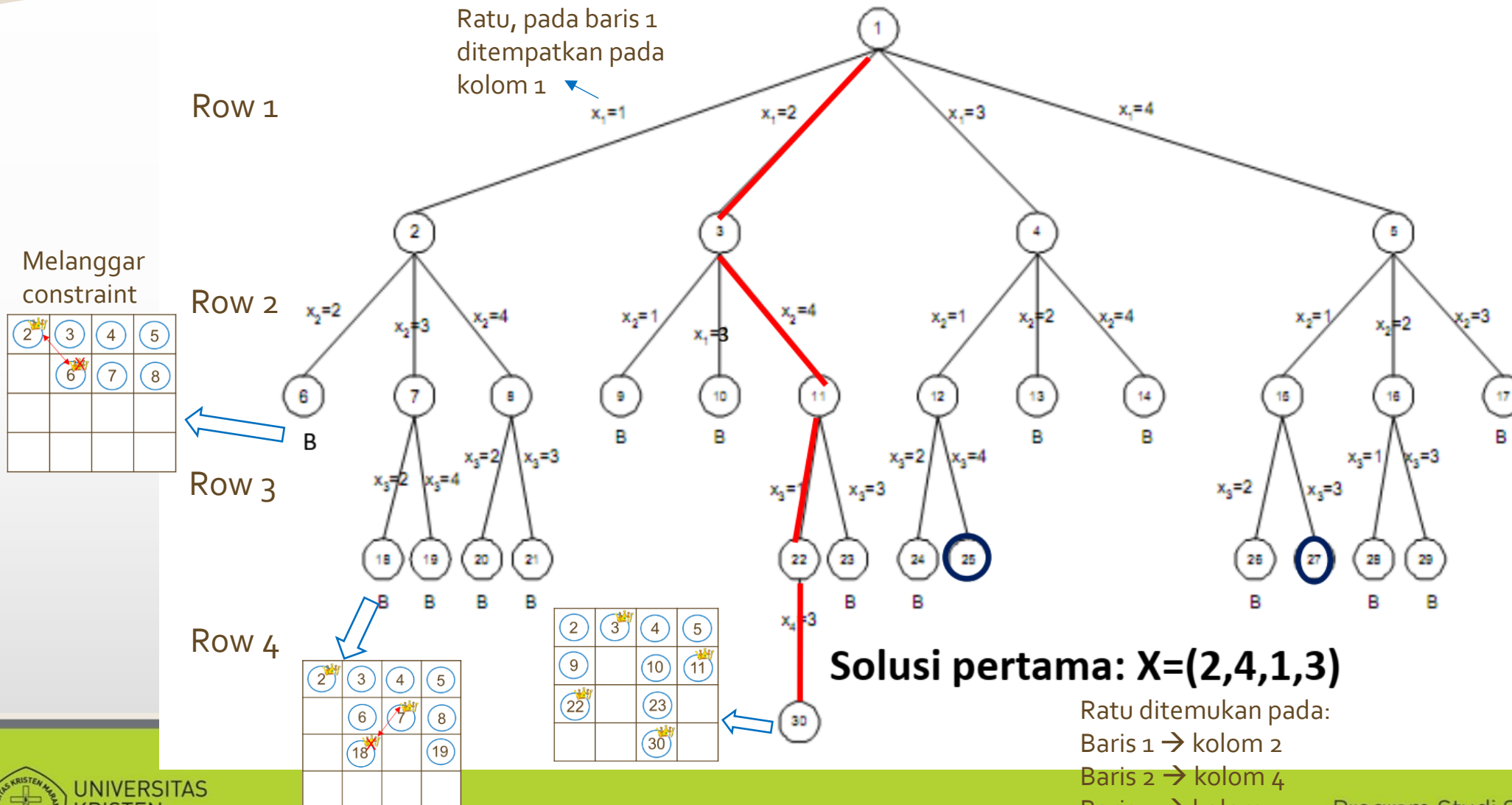
1. Masukkan simpul akar ke dalam *antrian Q*. Jika simpul akar adalah simpul solusi (goal node), maka solusi telah ditemukan. Jika hanya satu solusi yang diinginkan, maka stop.
2. Jika *Q* kosong, Stop.
3. Jika *Q* tidak kosong, pilih dari antrian *Q* simpul *i* yang mempunyai nilai 'cost' $\hat{c}(i)$ paling kecil. Jika terdapat beberapa simpul *i* yang memenuhi, pilih satu secara sembarang.
4. Jika simpul *i* adalah simpul solusi, berarti solusi sudah ditemukan. Jika satu solusi yang diinginkan, maka stop. Pada persoalan optimasi dengan pendekatan least cost search, periksa cost semua simpul hidup. Jika cost nya lebih besar dari cost simpul solusi, maka matikan simpul tersebut.
5. Jika simpul *i* bukan simpul solusi, maka bangkitkan semua anak-anaknya. Jika *i* tidak mempunyai anak, kembali ke langkah 2.
6. Untuk setiap anak *j* dari simpul *i*, hitung $\hat{c}(j)$, dan masukkan semua anak-anak tersebut ke dalam *Q*.
7. Kembali ke langkah 2.

Persoalan N-Ratu



- Diberikan sebuah papan catur yang berukuran $N \times N$ dan delapan buah ratu.
- Bagaimanakah menempatkan N buah ratu (Q) itu pada petak-petak papan catur sedemikian sehingga tidak ada dua ratu atau lebih yang terletak
 - Pada satu baris yang sama, atau
 - pada satu kolom yang sama, atau
 - pada satu diagonal yang sama ?

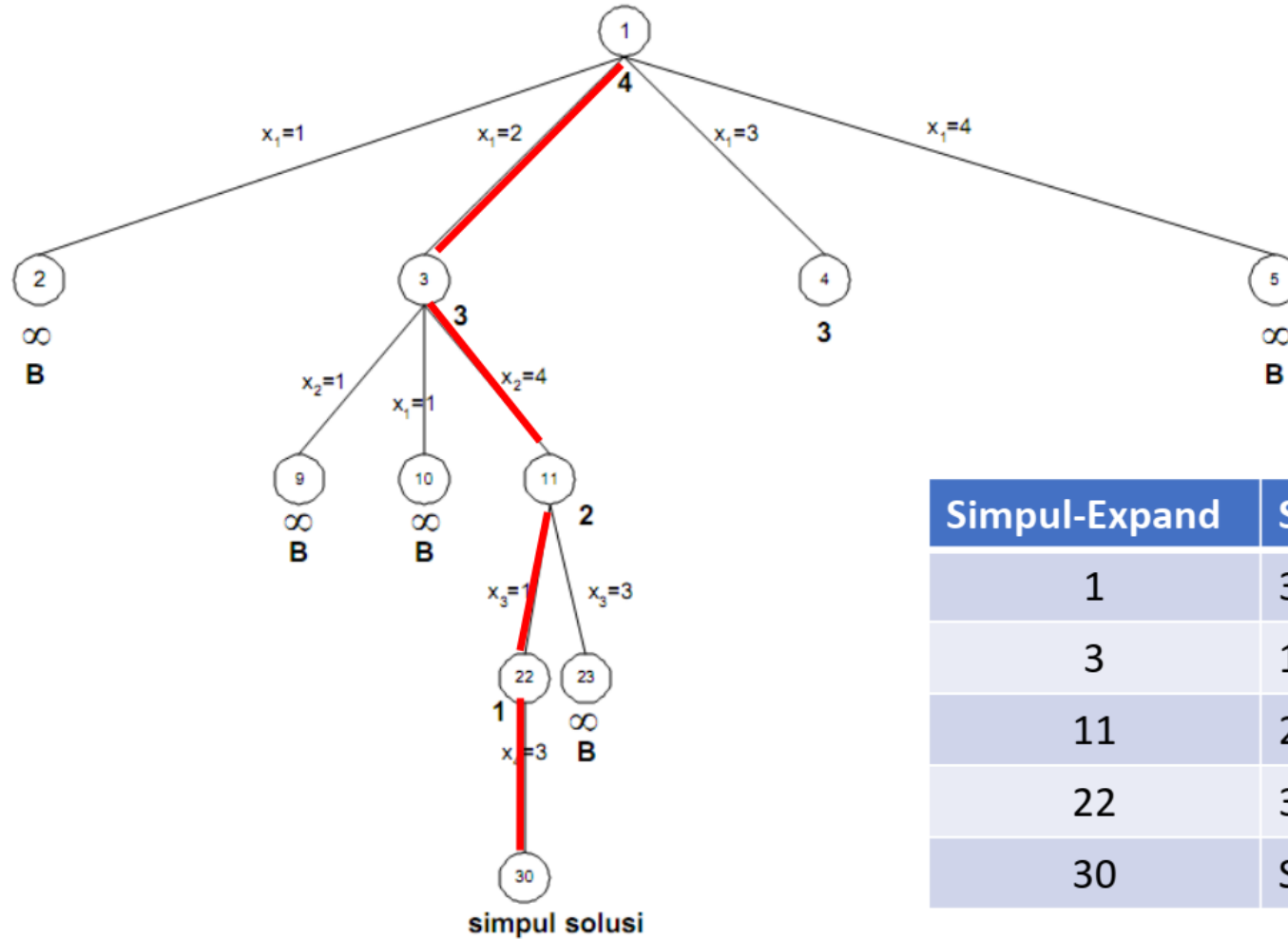
Persoalan 4-Ratu dengan BFS-dengan pembatasan (FIFO-Branch and Bound)



Solusi 4-Ratu

- 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]:
 - jumlah simpul dalam subpohon X yang perlu dibangkitkan sebelum simpul solusi ditemukan, atau
 - panjang lintasan dari simpul X ke simpul solusi terdekat (dalam subpohon X ybs)

Solusi Branch & Bound utk 4 Ratu

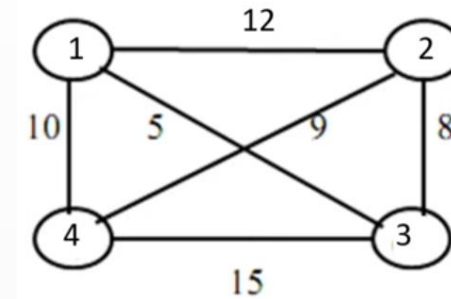
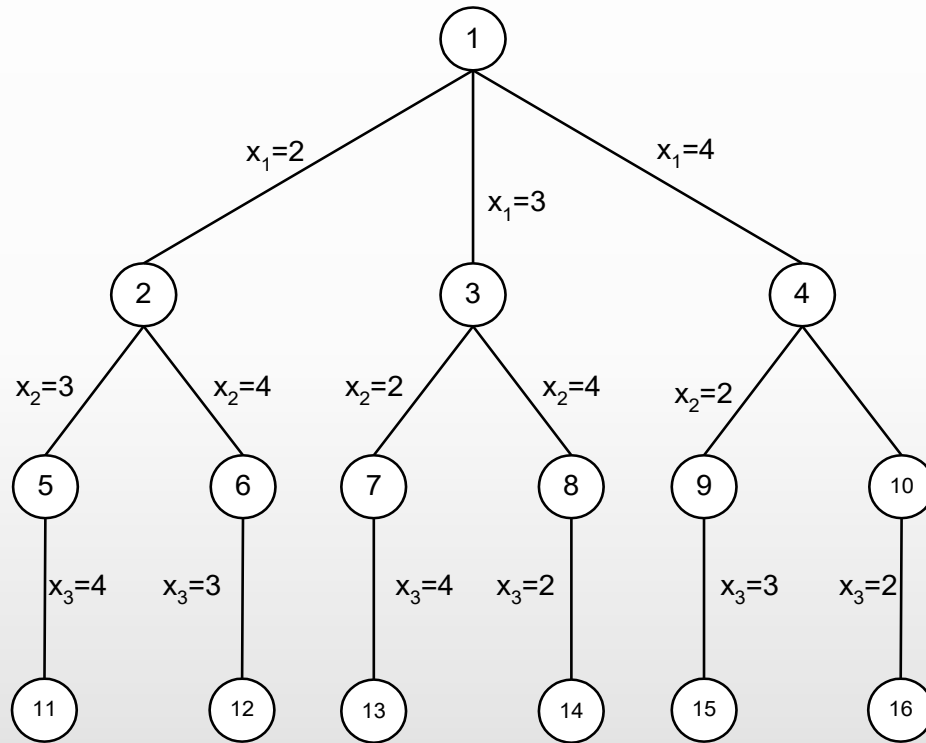


Simpul-Expand	Simpul Hidup
1	3,4,2,5
3	11,4,2,5,9,10
11	22,4,2,5,9,10,23
22	30,4,2,5,9,10,23
30	Solusi ketemu

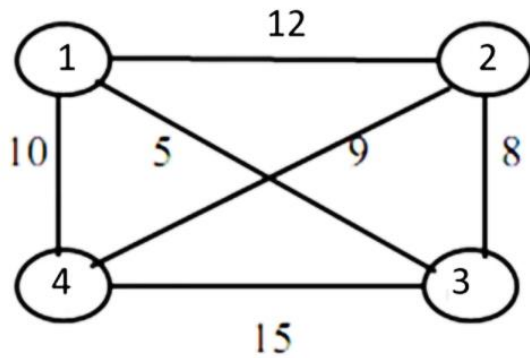
Priority queue

13

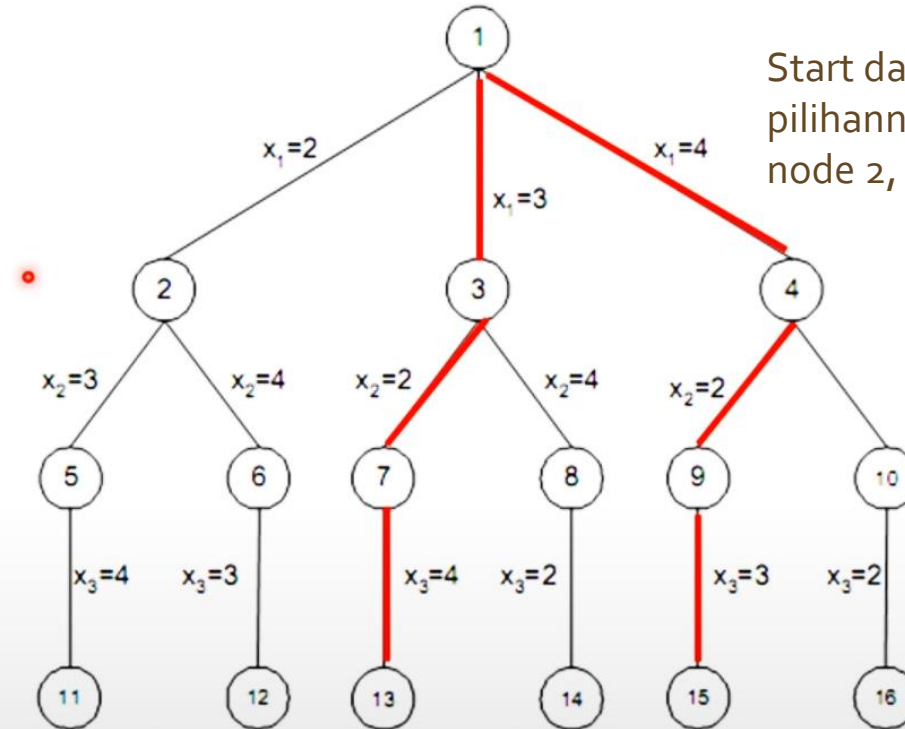
Pohon ruang status utk persoalan TSP dengan 4 simpul



Pohon ruang status utk persoalan TSP dengan 4 simpul



Simpul awal perjalanan = 1



Start dari node 1, kemudian pilihannya adalah menuju node 2, 3, atau 4

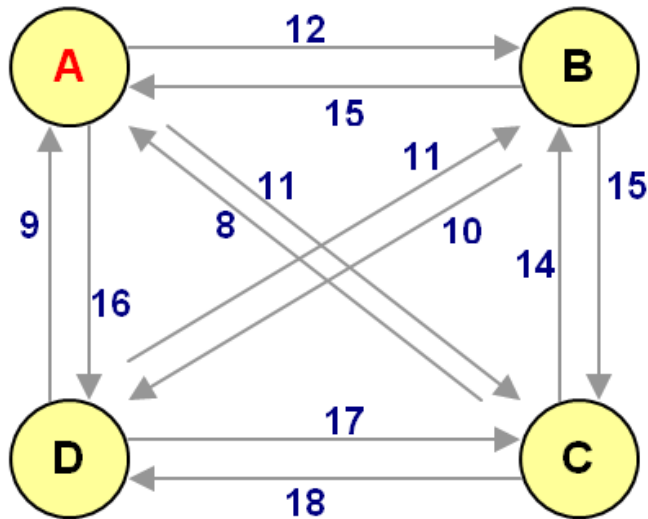
Solusi: 1-3-2-4-1 atau 1-4-2-3-1
Bobot = 5+8+9+10 = 32

Solusi B&B utk TSP – Reduced Matrix

- Ongkos atau nilai batas untuk setiap simpul dihitung dengan menggunakan matriks ongkos-tereduksi (*reduced cost matrix*) dari graf G .
- Sebuah matriks dikatakan tereduksi jika
 - setiap kolom dan barisnya mengandung **paling sedikit satu buah nol** dan **semua elemen lainnya non-negatif**.
 - Caranya :
 - Untuk setiap baris, cari nilai terkecil, nyatakan dengan $c(i)$. Kurangi semua nilai di baris itu dengan $c(i)$.
 - Untuk setiap kolom, cari nilai terkecil, nyatakan dengan $c(j)$. Kurangi semua nilai di kolom itu dengan $c(j)$.

Contoh persoalan

- TSP dengan $n = 4$

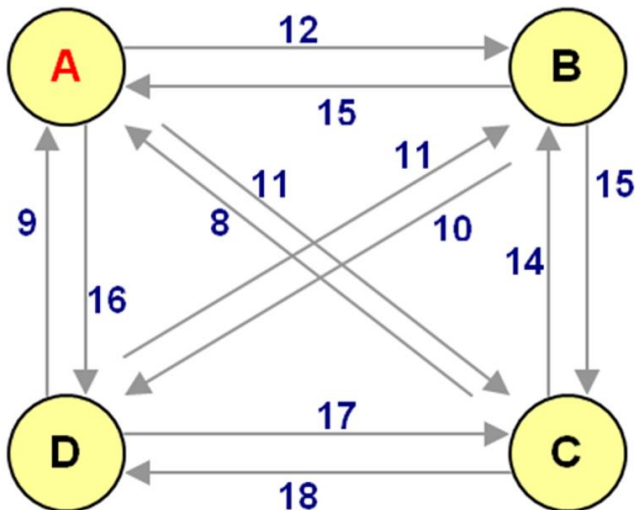


∞	12	11	16
15	∞	15	10
8	14	∞	18
9	11	17	∞

- Cost untuk setiap simpul pada pohon ruang status menyatakan nilai batas bawah (*lower bound*) ongkos mencapai simpul solusi dari simpul tersebut.
- Cost setiap simpul dapat dihitung secara heuristic berdasarkan:
 1. Matriks ongkos-tereduksi (*reduced cost matrix*) dari graf
 2. Bobot minimum tur lengkap

Reduksi Matriks

Sebuah matriks dikatakan tereduksi jika setiap kolom dan barisnya mengandung paling sedikit satu buah nol dan semua elemen lainnya non-negative.



∞	12	11	16
15	∞	15	10
8	14	∞	18
9	11	17	∞



∞	12	11	16
15	∞	15	10
8	14	∞	18
9	11	17	∞

R=11



∞	1	0	5
15	∞	15	10
8	14	∞	18
9	11	17	∞

R=11+10



∞	1	0	5
5	∞	5	0
8	14	∞	18
9	11	17	∞

R=21+8

R₃ - 8

∞	1	0	5
5	∞	5	0
0	6	∞	10
9	11	17	∞

R=29+9



∞	1	0	5
5	∞	5	0
0	6	∞	10
0	2	8	∞

R=38



∞	1	0	5
5	∞	5	0
0	6	∞	10
0	2	8	∞

R=38+0



∞	1	0	5
5	∞	5	0
0	6	∞	10
0	2	8	∞

R=38+1

C₂ - 1

∞	0	0	5
5	∞	5	0
0	5	∞	10
0	1	8	∞

R=39+0



∞	0	0	5
5	∞	5	0
0	5	∞	10
0	1	8	∞

R=39+0



∞	0	0	5
5	∞	5	0
0	5	∞	10
0	1	8	∞

R=39

Hitung Nilai Batas

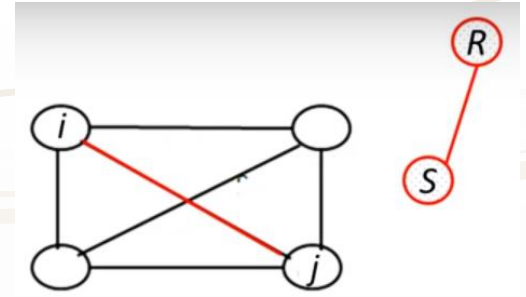
- Jumlah total elemen pengurang dari semua baris dan kolom menjadi batas bawah (*lower bound*) dari tur dengan total bobot minimum. Nilai ini digunakan sebagai nilai batas untuk simpul akar pada pohon ruang status.
- Total jumlah semua pengurang =
 - $(11 + 10 + 8 + 9 + 1) = 39$.
- Nilai $R = 39$ ini adalah nilai batas untuk simpul akar
- Pohon ruang status saat ini:

1

39

39 ini menjadi nilai cost untuk simpul akar. Sebuah tur dalam TSP akan memiliki lower bound sedikitnya 39.

Algoritma Branch & Bound utk TSP



- Misalkan:
 - Tur (perjalanan) dimulai dari simpul 1
 - A: matriks tereduksi untuk simpul akar R
 - S: anak dari simpul akar sehingga sisi (R, S) pada pohon ruang status berkoresponden dengan sisi (i, j) pada perjalanan.
- Jika S bukan simpul daun, maka matriks bobot tereduksi untuk simpul S dapat dihitung sebagai berikut:
 - ubah semua nilai pada baris i dan kolom j menjadi ∞ . Ini untuk mencegah agar tidak ada lintasan yang keluar dari simpul i atau masuk pada simpul j;
 - ubah $A(j, 1)$ menjadi ∞ . Ini untuk mencegah penggunaan sisi (j, 1);
 - reduksi kembali semua baris dan kolom pada matriks A kecuali untuk elemen ∞ .
- Jika r adalah total semua pengurang, maka nilai batas untuk simpul S adalah:

$$\hat{c}(S) = \hat{c}(R) + A(i, j) + r$$

Hasil reduksi ini menghasilkan matriks B.

Algoritma Branch & Bound utk TSP

- Secara umum, persamaan fungsi pembatas (*bounding function*) adalah:

$$\hat{c}(S) = \hat{c}(R) + A(i, j) + r$$

yang dalam hal ini,

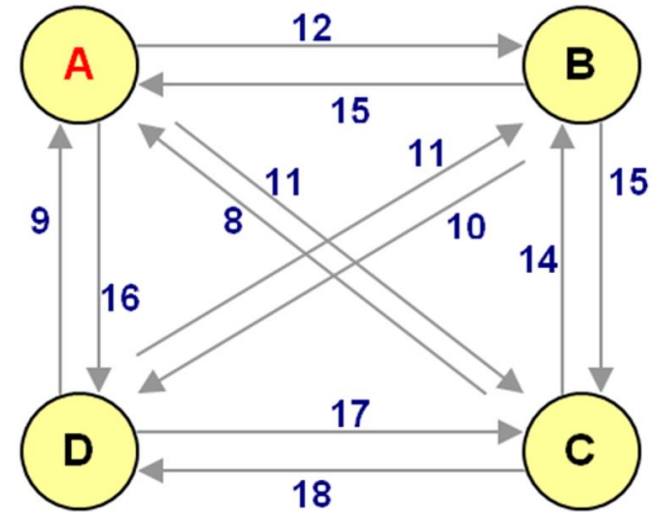
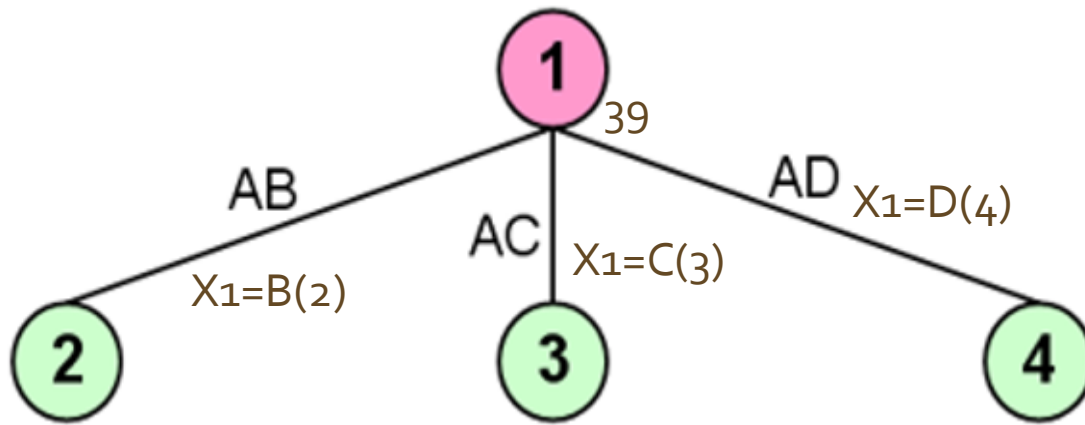
$\hat{c}(S)$ = bobot perjalanan minimum yang melalui simpul S (simpul di pohon ruang status)

$\hat{c}(R)$ = bobot perjalanan minimum yang melalui simpul R , yang dalam hal ini R adalah orangtua dari S .

$A(i, j)$ = bobot sisi (i, j) pada graf G yang berkoresponden dengan sisi (R, S) pada pohon ruang status.

r = jumlah semua pengurang pada proses memperoleh matriks tereduksi untuk simpul S .

Pembangkitan anak dari simpul akar



Kita sudah menghitung cost untuk simpul 1 (slide 15) $\rightarrow 39$
Selanjutnya, akan dihitung cost untuk simpul 2, 3, dan 4

B&B untuk TSP

- Simpul 2, lintasan dalam graf : AB (baris 1 – kolom 2)
- Ubah baris 1 dan kolom 2 menjadi ∞
- Ubah $M[2,1]$ menjadi ∞ , untuk mencegah penggunaan sisi (2,1)
- Reduksi semua baris dan kolom matriks

∞	0	0	5
5	∞	5	0
0	5	∞	10
0	1	8	∞

RCM Node 1
b=39



A	baris	1		
B	kolom	2		
	b	44	$=39+0+5$	
		∞	∞	∞
		∞	∞	5
		0	∞	10
		0	∞	8
			-5	
		∞	∞	∞
		∞	∞	0
		0	∞	10
		0	∞	3

- Simpul 3, lintasan dalam graf :
AC (baris 1 – kolom 3)

A	baris	1			
C	kolom	3			
	b	44	$=39+0+5$		
		∞	∞	∞	∞
		5	∞	∞	0
		∞	5	∞	10
		0	1	∞	∞
		∞	∞	∞	∞
		5	∞	∞	0
		∞	0	∞	5
		0	1	∞	∞

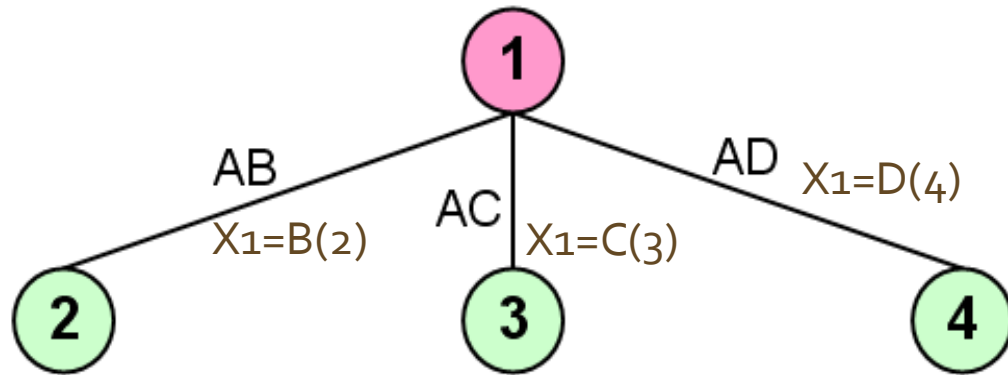
∞	0	0	5
5	∞	5	0
0	5	∞	10
0	1	8	∞

RCM Node 1
b=39

- Simpul 4, lintasan dalam graf :
AD (baris 1 – kolom 4)

A	baris	1			
D	kolom	4			
	b	50	$=39+5+6$		
		∞	∞	∞	∞
		5	∞	5	∞
		0	5	∞	∞
		∞	1	8	∞
		∞	∞	∞	∞
		0	∞	0	∞
		0	5	∞	∞
		∞	0	7	∞

Pohon ruang solusi tahap awal



∞	0	0	5
5	∞	5	0
0	5	∞	10
0	1	8	∞

RCM Node 1
 $b=39$

∞	∞	∞	∞
∞	∞	0	0
0	∞	∞	10
0	∞	3	∞

RCM Node 2
 $R=5$
 $b=39+0+5=44$

∞	∞	∞	∞
5	∞	∞	0
∞	0	∞	5
0	1	∞	∞

RCM Node 3
 $R=5$
 $b=39+0+5=44$

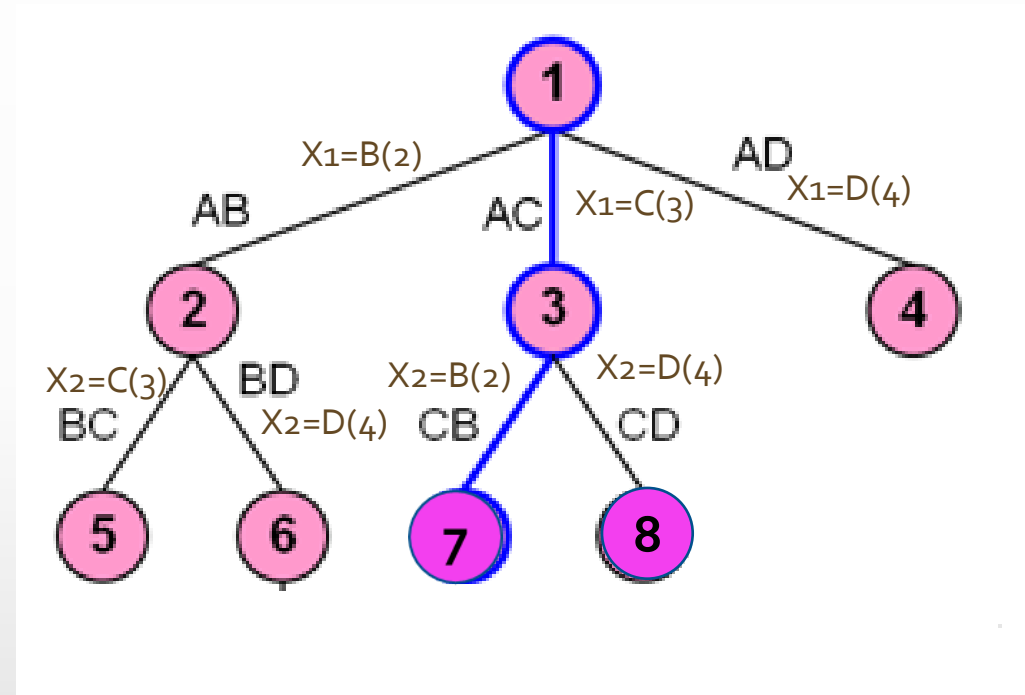
∞	∞	∞	∞
0	∞	0	∞
0	5	∞	∞
∞	0	7	∞

RCM Node 4
 $R=6$
 $b=39+5+6=50$

- Simpul yang mempunyai cost terkecil adalah simpul 2 dan simpul 3
- Simpul yang diexpand
 - Simpul 2
 - Simpul 3

Pembangkitan simpul berikutnya

- Anak simpul 2
 - Simpul 5 : BC
 - Simpul 6 : BD
- Anak simpul 3
 - Simpul 7 : CB
 - Simpul 8 : CD



- Simpul 5, lintasan dalam graf :
BC (baris 2 – kolom 3)

B	baris	2		
C	kolom	3		
b		54	$=44+0+10$	
		∞	∞	∞
		∞	∞	∞
		∞	∞	10
		0	∞	∞
		∞	∞	∞
		∞	∞	∞
		∞	∞	0
		0	∞	∞

∞	∞	∞	∞
∞	∞	0	0
0	∞	∞	10
0	∞	3	∞

- Simpul 6, lintasan dalam graf :
BD (baris 2 – kolom 4)

B	baris	2		
D	kolom	4		
b		47	$=44+0+3$	
		∞	∞	∞
		∞	∞	∞
		0	∞	∞
		∞	∞	3
		∞	∞	∞
		∞	∞	∞
		0	∞	∞
		∞	∞	0

- Simpul 7, lintasan dalam graf :
CB (baris 3 – kolom 2)

C baris	3			
B kolom	2			
b	44 min			
	∞	∞	∞	∞
	∞	∞	∞	0
	∞	∞	∞	∞
	0	∞	∞	∞

∞	∞	∞	∞
5	∞	∞	0
∞	0	∞	5
0	1	∞	∞

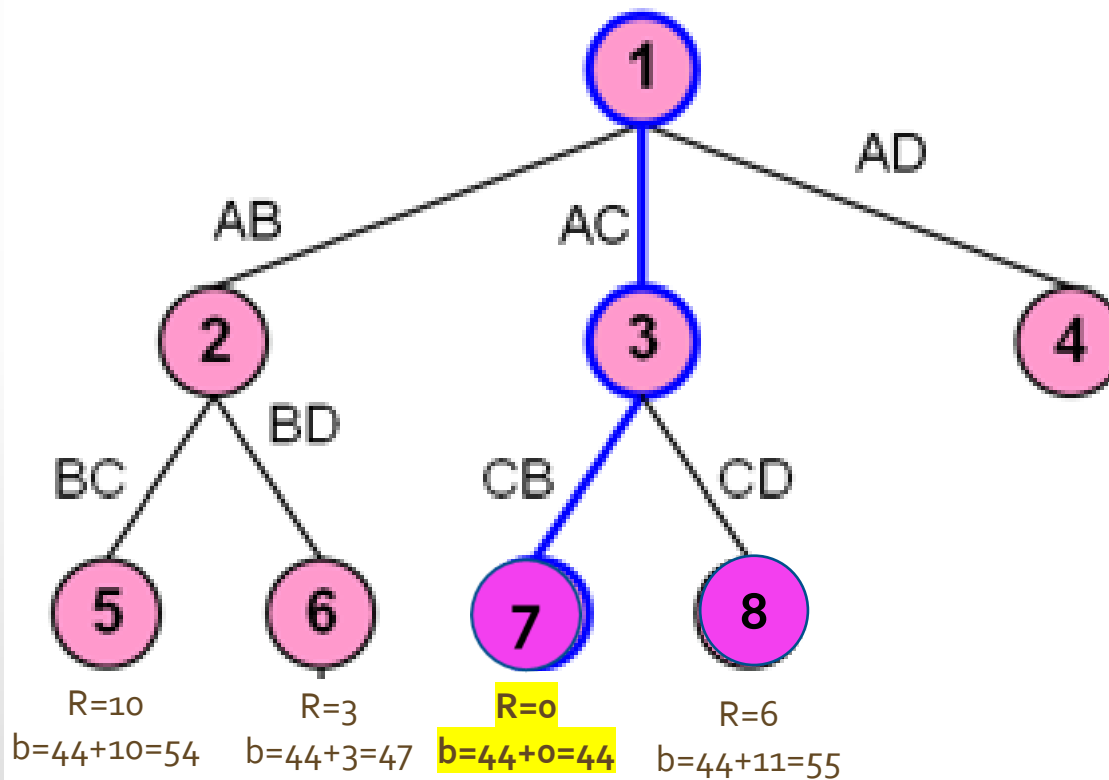
- Simpul 8, lintasan dalam graf :
CD (baris 3 – kolom 4)

C	baris	3			
D	kolom	4			
	b	55 = 44 + 5 + 6			
	∞	∞	∞	∞	
	5	∞	∞	∞	-5
	∞	∞	∞	∞	
	∞	1	∞	∞	-1
	∞	∞	∞	∞	
	0	∞	∞	∞	
	∞	∞	∞	∞	
	∞	0	∞	∞	

Pohon ruang solusi

∞	∞	∞	∞
∞	∞	0	0
0	∞	∞	10
0	∞	3	∞

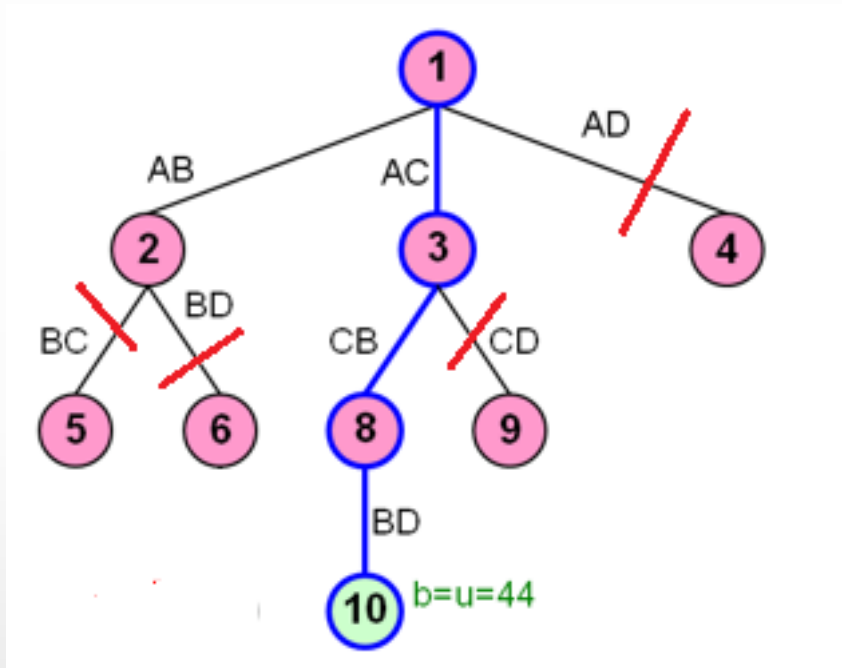
RCM Node 2
R=5
 $b=39+0+5=44$



∞	∞	∞	∞
5	∞	∞	0
∞	0	∞	5
0	1	∞	∞

RCM Node 3
R=5
 $b=39+0+5=44$

SOLUSI B&B utk TSP



Solusi TSP

Lintasan yang dihasilkan adalah

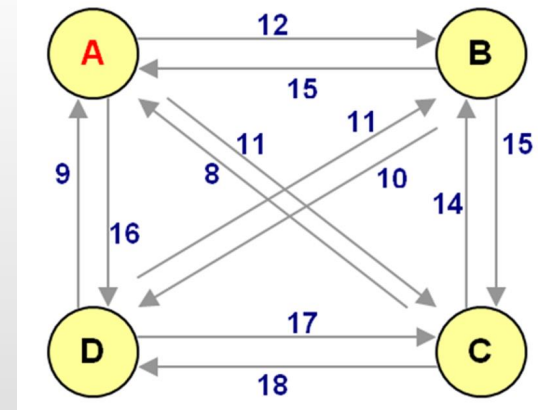
$$AC = 11$$

$$CB = 14$$

$$BD = 10$$

$$DA = 9$$

$$\text{Total lintasan} = 44$$



LATIHAN

- Diketahui Cost Matrix sebuah Graph TSP sebagai berikut :

∞	16	8	∞	12
10	∞	15	12	10
8	14	∞	10	8
12	17	10	∞	11
15	8	∞	18	∞

Kelas A

∞	12	7	∞	12
10	∞	15	17	10
8	16	∞	10	8
12	15	10	∞	11
11	8	∞	16	∞

Kelas B

- Temukan lintasan TSP terpendek dengan menggunakan Branch and Bound !

Referensi

- Anany Levitin, Introduction to The Design and Analysis of Algorithms, Pearson, 2012.
- S.Dasgupta, et al. Algorithms, 2006.
- Cormen, et al. Algorithms, MGH, 2009.
- Strategi Algoritma – Rinaldi Munir (ITB) :
 - <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/stmik.htm>
- Khan Academy, Computer Science: Algorithm,
<https://www.khanacademy.org/computing/computer-science/algorithms>