BAB III

BIAYA MINIMALPADA JARINGAN TRANSPORTASI

Untuk memudahkan dalam menentukan lintasan terpendek pada suatu jaringan transportasi, maka bentuk lintasan jalan raya yang digunakan, secara grafis dapat dinyatakan dalam bentuk graph berarah. Berikut ini akan dijelaskan pembentukan matrik tetangga dari graph berarah, pembentukan matrik biaya dan tutupan menghantar (*transitif closur*) yang digunakan untuk mendapatkan matrik lintasan.

3.1. Matrik Tetangga

Matrik tetangga adalah matrik yang elemennya menunjukkan ada tidaknya suatu hubungan antara verteks v_i dengan verteks v_j melalui suatu rusuk (garis) berarah pada suatu graph berarah. Hal ini sesuai dengan pengertian tetangga dari graph berarah, yaitu tetangga suatu verteks, misalnya v_i , dan ada verteks lain, misalnya v_j, yang dihubungkan dengan rusuk yang berarah, yang arahnya dari verteks v_i ke verteks v_j. Dalam gambar 3.1., verteks 2 dikatakan bertetangga dengan verteks 3, tetapi tidak untuk sebaliknya.

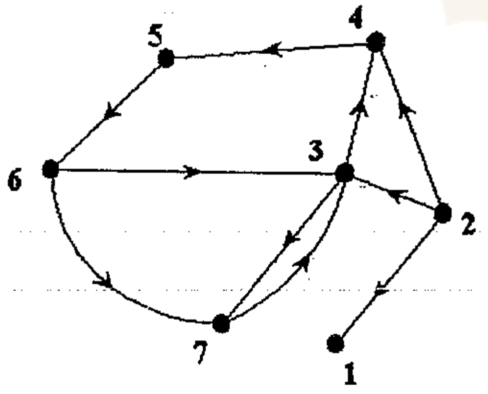
Untuk membentuk matrik tetangga, langkah pertama yang harus dilakukan adalah memberi nama untuk setiap verteks yang ada, misalnya dari nomor 1 sampai dengan n, dengan n adalah banyaknya verteks yang ada. Setelah langkah tersebut dilakukan maka matrik tetangga dapat didefinisikan sebagai matrik berukuran $n \times n$ yang nilai elemen dapat didefinisikan sebagai berikut :

$$a_{ij} = \begin{cases} 1 & \text{jika } v_i \text{ bertetangga dengan } v_j \\ 0 & \text{jika } v_i \text{ tidak bertetangga dengan } v_j \end{cases} \qquad (3.1)$$

Dengan melihat definisi di atas dapat dikatakan nilai elemen dari matrik tetangga adalah 1 atau 0, yang sering disebut matrik bit atau matrik Boolean.

Matrik tetangga dari sembarang graph berarah G=(V,E) tergantung pada cara memasangkan verteks dan arah aliran antara satu verteks dengan verteks lainnya. Setiap pasangan yang berbeda, atau setiap arah aliran yang berbeda, akan menghasilkan matrik tetangga yang berbeda. Meskipun demikian, dapat dilihat bahwa matrik-matrik yang dihasilkan dari cara pengurutan yang berbeda dapat diperoleh dari matrik yang lain dengan cara menukar baris atau kolom.

Dengan mengambil contoh graph berarah pada gambar 3.1., maka matrik tetangga untuk graph berarah tersebut adalah:



$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Gambar 3.1. graph berarah dan matrik tetangganya

Dari matrik tetangga diatas dapat dilihat bahwa banyaknya elemen yang bernilai $\neq 0$ pada matrik A adalah sama dengan banyaknya rusuk pada graph berarah pada gambar 3.1.

Dapat ditentukan suatu cara untuk menentukan jumlah dari semua kemungkinan panjang lintasan atau hubungan r - langkah (r = 1,2, ...) dari verteks v_i ke verteks v_j pada sembarang graph berarah , (termasuk kasus dimana v_i dan v_j adalah verteks yang sama). Jumlah dari hubungan 1- langkah dari v_i ke v_j secara sederhana adalah a_{ij}. Berarti terdapat kemungkinan salah satu nol atau satu hubungan 1 - langkah dari v_i ke v_j, tergantung nilai dari a_{ij} yaitu bernilai nol atau satu.

Untuk mendapatkan hubungan 2 - langkah, dapat ditentukan dengan matrik tetangga bujursangkar A^2, dimana A adalah matrik tetangga dari graph G = (V,E). Jika $a_{ij}^{(2)}$ adalah elemen ke-(i,j) dari A^2 maka didapat

$$a_{ij}^{(2)} = a_{i1}a_{1j} + a_{i2}a_{2j} + \ldots + a_{in}a_{nj} \tag{3.2}$$

Jika $a_{i1} = a_{1j} = 1$, maka terdapat hubungan 2-langkah $v_i \rightarrow v_1 \rightarrow v_j$ dari verteks v_i ke v_j, tetapi jika salah satu a_{i1} atau a_{1j} adalah nol maka hubungan 2-langkah tidak ada. Pada lintasan $v_i \rightarrow v_1 \rightarrow v_j$ terdapat hubungan 2-langkah jika dan hanya jika $a_{i1} \cdot a_{1j} = 1$. Dengan cara yang sama, untuk k = 1,2, ..., n, $v_i \rightarrow v_k \rightarrow v_j$ terdapat hubungan 2 – langkah dari v_i ke v_j jika dan hanya jika suku $a_{ik}a_{kj}$ dari persamaan (3.2) bernilai 1, dan nilai – nilai suku lainnya adalah 0. Dari hal ini, sisi kanan dari persamaan (3.2) adalah jumlah total dari dua hubungan 2-langkah dari v_i ke v_j . Dengan cara yang sama dapat ditentukan jumlah dari 3,4, ... , n langkah dari v_i ke v_j. Sehingga secara umum didapatkan hasil sebagai berikut:

Teorema 3.1.

Misalkan A adalah matrik tetangga dari graph berarah, dan $a_{ij}^{(r)}$ adalah elemen ke-(i,j) dari A^r, maka $a_{ij}^{(r)}$ menunjukkan jumlah lintasan dari v_i ke v_j.

Sebagai contoh, jika matrik A (matrik dari gambar 3.1) dikuadratkan (pangkat 2) maka elemen ke-(i,j) yang mempunyai nilai 1 dari matrik A^2, menunjukkan bahwa banyaknya lintasan dari verteks v_i ke verteks v_j sebanyak 2 lintasan. Dengan cara yang sama, jika matrik A dipangkatkan tiga, maka elemen ke-(i,j) yang mempunyai nilai 1 dari matrik A^3, menunjukkan bahwa banyaknya lintasan dari verteks v_i ke verteks v_j sebanyak 3 lintasan, dan seterusnya. Dengan menggunakan contoh matrik tetangga A diatas, pangkat 2, pangkat 3 dan pangkat 4 dari matrik A tersebut adalah:

$$A^2 = \begin{array}{cc} & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \end{array} \qquad A^3 = \begin{array}{cc} & \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Elemen ke-(3,4) dari matrik A^3 yang merupakan pangkat 3 dari matrik tetangga A (gambar 3.1) bernilai 1, yang menunjukkan verteks 4 dapat dijangkau dari verteks 3 dengan 3 lintasan, yaitu $v_3 \rightarrow v_7 \rightarrow v_3 \rightarrow v_4$. Pada matrik A^4, elemen ke-(3,3) nya bernilai 2, artinya vertek 3 dapat dijangkau dari verteks 3 dengan 2 cara, yang mempunyai panjang lintasan 4 lintasan (rusuk), cara tersebut sebagai berikut: $v_3 \rightarrow v_4 \rightarrow v_5 \rightarrow v_6 \rightarrow v_3$ dan $v_3 \rightarrow v_7 \rightarrow v_3 \rightarrow v_7 \rightarrow v_3$. Analog untuk verteks lain.

$$A^4 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{array} \begin{array}{c} \begin{array}{ccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 2 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Dengan melihat pada contoh diatas, maka terdapat 13 lintasan dengan panjang 2, terdapat 17 lintasan dengan panjang 3, dan terdapat 23 lintasan dengan panjang 4.

3.2. Pembentukan Matrik Biaya

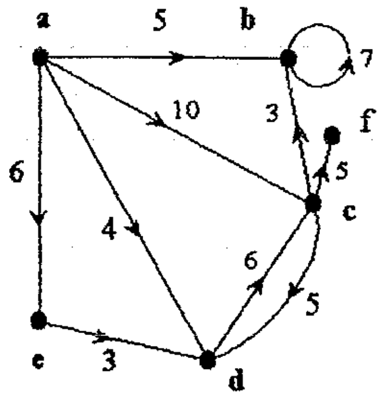
Penyajian graph seringkali tidak cukup hanya dengan menggunakan matrik tetangga dan matrik lintasan. Pada persoalan untuk mendapatkan biaya yang minimal pada jaringan transportasi sering harus dihitung berapa besar biaya yang harus dikeluarkan untuk berjalan dari satu kota ke kota lainnya.

Untuk menghitung biaya minimal, maka graph perlu dinyatakan dengan matrik bobot (*weight matrix*) yang nilai elemennya menunjukkan biaya yang harus ditanggung atau dikeluarkan untuk berpindah dari satu kota ke kota lainnya.

Matrik biaya pada dasarnya hampir sama dengan matrik tetangga, yaitu sebuah matrik yang mempunyai ukuran $n \times n$ (dengan n adalah banyaknya verteks) dan didefinisikan sebagai berikut:

$$w_{ij} = \begin{cases} \textit{biaya} & \text{jika ada lintasan antara } v_i \text{ dengan } v_j \\ 0 & \text{jika tidak ada lintasan antara } v_i \text{ dengan } v_j \end{cases} \qquad (3.3)$$

dengan *biaya* adalah suatu besaran yang menunjukkan biaya yang harus dikeluarkan (besarnya biaya yang harus ditanggung) jika harus melintasi dari kota v_i ke kota v_j. Gambar 3.2. berikut ini menyajikan contoh sebuah graph dan matrik biayanya.



$$W = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \end{array} \begin{array}{c} \begin{array}{cccccc} a & b & c & d & e & f \end{array} \\ \begin{bmatrix} 0 & 5 & 10 & 4 & 6 & 0 \\ 0 & 7 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 5 & 0 & 5 \\ 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Gambar 3.2 Contoh graph berarah dan matrik biayanya

3.3 Tutupan Menghantar (*Transitif Closur*)

Seringkali suatu verteks tidak dihubungkan secara langsung dengan verteks lainnya, sehingga untuk dapat menentukan lintasan terpendek yang melewati verteks-verteks tersebut tidak dapat dilakukan dengan efesien. Dengan adanya sifat transitif closur pada relasi biner, permasalahan untuk mendapatkan suatu lintasan baru pada graph berarah G=(V,E) tersebut dapat ditentukan dengan lebih efesien.

Suatu permasalahan dalam graph berarah untuk mendapatkan suatu lintasan terpendek diantara beberapa verteks adalah menentukan keberadaaan lintasan yang sesuai dari verteks v_i ke verteks v_j untuk setiap pasangan verteks $(i,j) \in G$

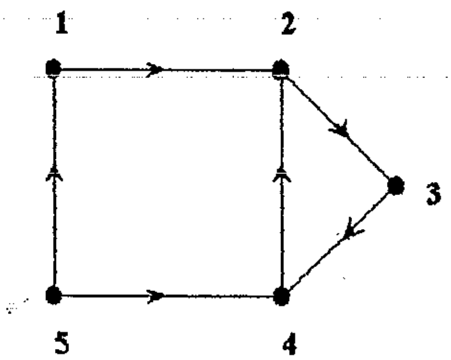
Sifat transitif closur pada lintasan dapat dikembangkan dari matrik tetangga, yaitu dengan menentukan matrik hubungan $A^* = (a_{ij}^*)$ yang didefinisikan dengan:

$$a_{ij}^* = \begin{cases} 1 & \text{jika ada lintasan dalam } G=(V,E) \text{ dari verteks i ke verteks j} \\ 0 & \text{jikatidak ada lintasan} \end{cases} \qquad (3.4)$$

Dari definisi diatas dapat dilihat bahwa matrik A^* juga merupakan matrik tetangga untuk graph $G^* = (V,E^*)$ dengan E^* merupakan transitif closur dari relasi biner E. Hal ini dimaksudkan membentuk graph berarah G^* baru yang berukuran n-verteks, dengan cara menambahkan rusuk-(i,j) yang berarah dari verteks v_i ke verteks v_j ke dalam graph G asal, jika dan hanya jika ada lintasan (dengan panjang 2,3 … n-1) dari dari verteks v_i ke verteks $v_j \in G$. Dengan kata lain transitif closur adalah

$$A(G^*) = R(G) \qquad (3.5)$$

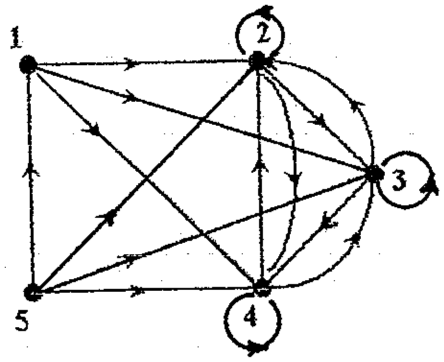
dengan $A(G^*)$ adalah matrik tetangga dari graph G dan R(G) adalah matrik keterjangkauan dari G, yaitu matrik yang elemen ke-(i,j)nya menunjukkan verteks v_j dapat dijangkau dari verteks v_i.



(a) Graph G

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

(c) Matrik tetangga A dari graph G



$$
\begin{array}{c|ccccc|}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 0 & 1 & 1 & 1 & 0 \\
2 & 0 & 1 & 1 & 1 & 0 \\
3 & 0 & 1 & 1 & 1 & 0 \\
4 & 0 & 1 & 1 & 1 & 0 \\
5 & 1 & 1 & 1 & 1 & 0
\end{array}
$$

(b) Graph transitif closur G^* (d) Matrik transitif closur G^*

Gambar 3.3 Menunjukkan (a)graph berarah G dan (b) transitif closur G^*
(c) Matrik tetangga dari graph G dan (d) matrik transitif Closur dari G^*

Matrik transitif closur A^* dapat ditentukan dari matrik tetangga A dengan mendefinisikan secara berurutan matrik $A^0 = (a_{ij}^{(0)})$, $A^1 = (a_{ij}^{(1)}),\ldots,A^n = (a_{ij}^{(n)})$ sebagai berikut:

$$a_{ij}^{(0)} = a_{ij}$$

$$a_{ij}^{(l)} = a_{ij}^{(l-1)} \vee (a_{ij}^{(l-1)} \wedge a_{ij}^{(l-1)}) \qquad (3.6)$$

Dinyatakan bahwa $A^* = A^n$, akan dibuktikan dengan cara induksi

$a_{ij}^{(l)} = 1$ jika dan hanya jika ada lintasan dari $v_i \in V$ ke $v_j \in V$ dengan verteks-verteks lanjutan pada lintasan dipilih hanya dari $\{1,2,\ldots,l\} \subseteq V$. Untuk $l = 0$ hal ini jelas. Jika benar untuk l-1 maka $a_{ij}^{(l-1)} = 1$ (terdapat suatu lintasan dari v_i ke v_j hanya melewati verteks (1,2,…,l-1), sehingga benar juga untuk untuk l.

Persamaan (3.6) diatas dapat digunakan untuk mentransformasikan matrik A menjadi matrik A^*. Selain cara diatas, persamaan (3.6) diatas dapat ditinjau dengan cara lainnya.

Perhatikan bahwa $A^{(l-1)}$ dapat ditranformasikan menjadi $A^{(l)}$, dengan cara sebagai berikut: jika $a_{il}^{(l-1)} = 0$ maka didapat $a_{ij}^{(l)} = a_{ij}^{(l-1)}$; dan jika $a_{il}^{(l-1)} = 1$ akan diperoleh $a_{ij}^{(l)} = a_{il}^{(l-1)} \vee a_{lj}^{(l-1)}$.

Misalkan a_{i*} menyatakan baris ke – i dari matrik A sehingga diperoleh

$$a_{i*}^{(l)} = \begin{cases} a_{i*}^{(l-1)} & \text{jika } a_{il}^{(l-1)} = 0 \\ a_{i*}^{(l-1)} \vee a_{l*}^{(l-1)} & \text{jika } a_{il}^{(l-1)} = 1 \end{cases} \tag{3.7}$$

Untuk i ≠ l, maka nilai dari $a_{i*}^{(l-1)}$ dimasukkan ke dalam perhitungan hanya untuk $a_{i*}^{(l)}$ dan selanjutnya nilainya dapat diganti tanpa mempengaruhi perhitungan dari $a_{k*}^{(l)}$ untuk k≠i. Selanjutnya $a_{l*}^{(l-1)}$ digunakan untuk menghitung baris yang lainnya tetapi dengan adanya persamaan (3.6) diatas dapat diperoleh juga $a_{l*}^{(l)} = a_{l*}^{(l-1)}$.

3.3.1. Pembentukan Matrik Lintasan

Seperti halnya dengan matrik tetangga, maka matrik lintasan P (*path matrix*) adalah matrik berukuran $n \times n$ yang nilai elemennya dapat ditentukan berdasarkan persamaan (3.4).

Dengan memperhatikan bahwa antara verteks v_i ke verteks v_j ada kemungkinan terdapat lebih dari satu lintasan dengan panjang yang berbeda-beda, maka matrik lintasan dapat ditentukan dengan cara sebagai berikut. Dimisalkan bahwa terdapat lintasan antara verteks v_i ke verteks v_j. Hal ini berarti bahwa terdapat lintasan sederhana dari verteks v_i ke verteks v_j jika verteks v_i tidak sama dengan verteks v_j. Karena graph hanya mempunyai n verteks, lintasan sederhana tersebut pasti mempunyai panjang n - 1 atau kurang, atau ada lingkar yang panjangnya n atau kurang.

Dari subbab 3.1. tentang matrik tetangga, dapat diketahui bagaimana menentukan panjang lintasan 1,2,3, dan seterusnya. Matrik-matrik yang diperoleh dengan menggunakan teorema 3.1, selanjutnya dapat digunakan untuk mendapatkan matrik lintasan untuk semua pasanagan verteks, tanpa kendala terhadap panjang lintasan, yaitu dengan menggunakan persamaan (3.8).

$$M_r = A \vee A^2 \vee A^3 \vee A^4 \vee \ldots \vee A^r \qquad (3.8)$$

Dengan memperhatikan penjelasan diatas dapat dilihat hubungan antara matrik tetangga A dengan matrik lintasan P, yaitu bahwa jika A adalah matrik tetangga dan P adalah matrik lintasan dari sembarang graph, maka $p_{ij} = 1$ jika dan hanya jika elemen b_{ij} tidak sama dengan nol, dimana b_{ij} adalah elemen dari matrik

$$B_m = A + A^2 + A^3 + A^4 + \ldots + A^m \qquad (3.8a)$$

Dengan menggunakan matrik dari gambar 3.1 maka matrik B_4 dapat diperoleh dari persamaan (3.8a)dengan r = 4, sehingga $B_4 = A + A^2 + A^3 + A^4$. Matrik lintasan P dapat diperoleh dengan cara mengganti seluruh elemen dari matrik B_4 yang tidak sama dengan nol dengan nilai 1. Matrik B_4 dan matrik lintasan P dapat sajikan seperti terlihat berikut ini.

$$\mathbf{B_4} = \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 3 & 3 & 2 & 2 & 2 \\ 0 & 0 & 3 & 2 & 2 & 1 & 3 \\ 0 & 0 & 2 & 1 & 1 & 1 & 2 \\ 0 & 0 & 3 & 2 & 1 & 1 & 3 \\ 0 & 0 & 4 & 3 & 2 & 1 & 4 \\ 0 & 0 & 2 & 2 & 1 & 1 & 2 \end{bmatrix} \qquad \mathbf{P} = \begin{array}{c} \mathbf{1} \\ \mathbf{2} \\ \mathbf{3} \\ \mathbf{4} \\ \mathbf{5} \\ \mathbf{6} \\ \mathbf{7} \end{array} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$