

# Atstumo žemėlapių, skirtų trumpiausio kelio paieškai, atvaizdavimas panaudojant ketvirtainių medžių duomenų struktūras

Jonas Antanaitis

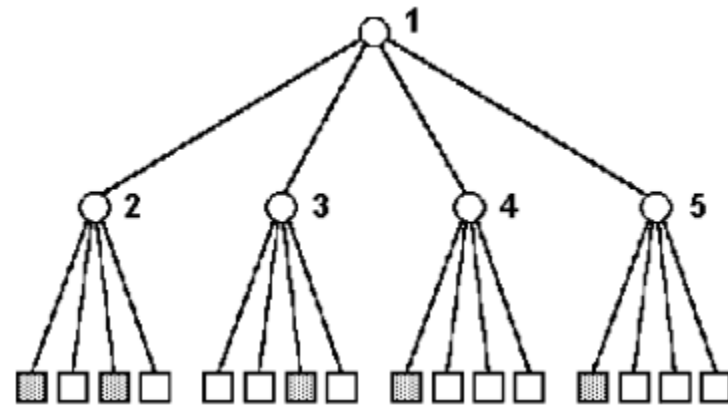
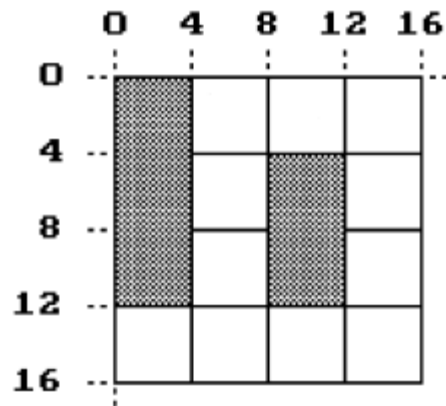
Kompiuterinis modeliavimas, 2 grupė

2014

# Turinys

- Įvadas
- Ketvirtainio medžio matrica
- Atstumo žemėlapiai
- Kaimynų suradimo algoritmas
- Kelio planavimas plokštumoje
- Išplėstiniai ketvirtainiai medžiai
- Svorio koeficientai
- Kelio planavimas erdvėje
- Išvados

# Ketvirtainio medžio matrica



# Ketvirtainio medžio matrica

- $L \times 4$  didžio matrica:

$$R = [P_1, P_2, P_3, \dots, P_L]^T,$$

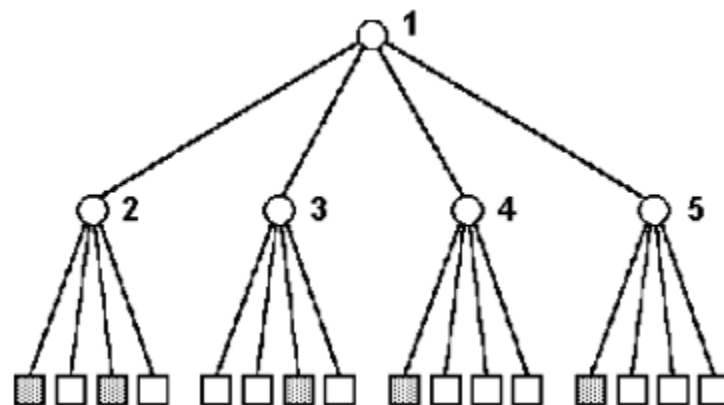
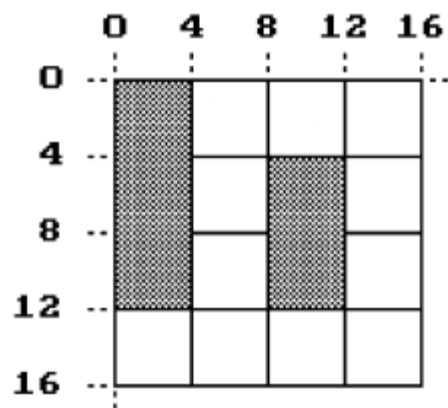
- Ryšis tarp  $k$  ir  $k-1$  lygio mazgų:

$$P_i = (Q_{i1}, Q_{i2}, Q_{i3}, Q_{i4}),$$

- Lapuose saugoma informacija:

$$Q_{im} = \begin{cases} -c \\ 0 \end{cases}$$

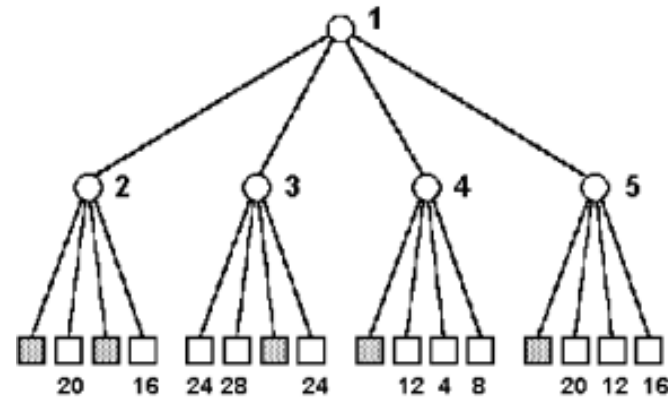
# Ketvirtainio medžio matrica



| $P_i/Q_{ij}$ | 0  | 1  | 2  | 3  |
|--------------|----|----|----|----|
| 1            | 2  | 3  | 4  | 5  |
| 2            | 0  | -1 | 0  | -1 |
| 3            | -1 | -1 | 0  | -1 |
| 4            | 0  | -1 | -1 | -1 |
| 5            | 0  | -1 | -1 | -1 |

# Atstumo žemėlapis atvaizduotas, ketvirtainiu medžiu

|    | 0 | 4  | 8  | 12 | 16 |
|----|---|----|----|----|----|
| 0  |   | 20 | 24 | 28 |    |
| 4  |   | 16 |    | 24 |    |
| 8  |   | 12 |    | 20 |    |
| 12 |   | 4  | 8  | 12 | 16 |
| 16 |   |    |    |    |    |



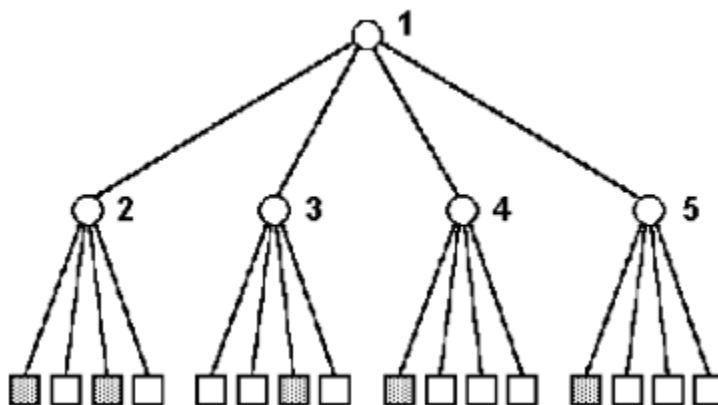
| $P_i/Q_j$ | 0   | 1   | 2   | 3   |
|-----------|-----|-----|-----|-----|
| 1         | 2   | 3   | 4   | 5   |
| 2         | 0   | -20 | 0   | -16 |
| 3         | -24 | -28 | 0   | -24 |
| 4         | 0   | -12 | -4  | -8  |
| 5         | 0   | -20 | -12 | -16 |

# Atstumų žemėlapiai. Jų užpildymas

|    | 0 | 4  | 8  | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 | 44 | 48 |
|----|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  |   |    |    |    |    |    |    |    |    |    |    |    |    |
| 4  | 8 |    | 16 |    | 24 |    | 32 |    | 1  |    | 1  |    |    |
| 8  |   | 16 | 20 | 20 | 24 | 28 | 1  | 1  | 1  | 1  |    | 1  |    |
| 12 |   |    | 20 |    |    |    |    |    |    |    |    |    |    |
| 16 |   | 24 | 24 |    |    |    |    |    |    |    |    | 1  |    |
| 20 |   |    | 28 |    |    |    |    |    |    |    |    |    |    |
| 24 |   | 32 | 1  |    |    |    |    |    |    |    |    |    |    |
| 28 |   |    | 1  |    |    |    | 1  |    | 1  |    | 1  |    |    |
| 32 |   |    |    |    |    |    |    |    |    |    |    |    |    |
| 36 |   | 1  | 1  |    |    |    | 1  |    |    |    |    |    | 1  |
| 40 |   |    | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |
| 44 |   | 1  |    | 1  |    | 1  |    | 1  |    |    |    |    |    |
| 48 |   |    |    |    |    |    |    |    |    |    |    |    |    |
| 52 |   |    | 1  |    |    | 1  |    |    |    |    |    |    |    |

|    | 0 | 4  | 8  | 12 | 16 | 20 | 24 | 28 | 32 | 36 | 40 | 44 | 48 |
|----|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 0  |   |    |    |    |    |    |    |    |    |    |    |    |    |
| 4  |   | 8  |    | 16 |    | 24 |    | 32 |    | 1  |    | 1  |    |
| 8  |   |    | 16 | 20 | 20 | 24 | 28 | 1  | 1  | 1  | 1  |    | 1  |
| 12 |   |    |    | 20 | 28 |    |    |    |    |    |    |    |    |
| 16 |   |    | 24 | 24 |    |    |    |    |    |    |    |    |    |
| 20 |   |    |    | 28 |    |    |    |    |    |    |    |    |    |
| 24 |   | 32 |    | 1  |    |    |    | 1  |    | 1  |    | 1  |    |
| 28 |   |    |    | 1  |    |    |    |    |    |    |    |    |    |
| 32 |   |    | 1  | 1  |    |    |    | 1  |    |    |    |    | 1  |
| 36 |   |    |    | 1  | 1  | 1  | 1  |    |    |    |    |    |    |
| 40 |   | 1  |    | 1  |    | 1  |    | 1  |    |    |    |    |    |
| 44 |   |    |    |    |    |    |    |    |    |    |    |    |    |
| 48 |   |    |    |    |    |    |    |    |    |    |    |    |    |
| 52 |   |    | 1  |    |    | 1  |    |    |    |    |    |    |    |

# Kaimynų ieškojimo algoritmas



$$A = \begin{pmatrix} x_0 y_0 \\ x_1 y_1 \\ \vdots \\ x_n y_n \end{pmatrix}$$

$$x_A = \sum_{i=0}^n x_i \times 2^{N-i-1},$$

$$y_A = \sum_{i=0}^n y_i \times 2^{N-i-1},$$

$$d_A = 2^{N-n-1}.$$

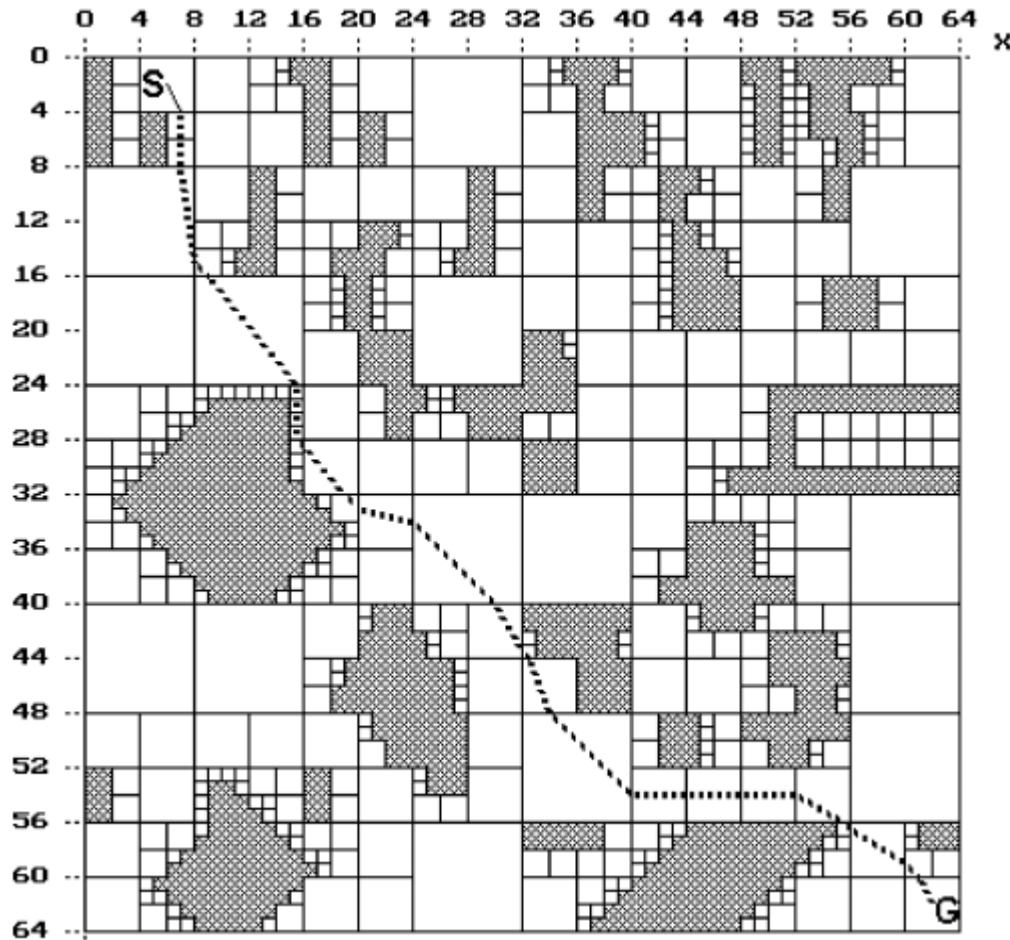


# Kelio planavimas plokštumoje

- Sudarius atstumo žemėlapyj naudojama greičiausio nusileidimo strategija.

|    | 0 | 4  | 8  | 12 | 16 |
|----|---|----|----|----|----|
| 0  |   | 20 | 24 | 28 |    |
| 4  |   | 16 |    | 24 |    |
| 8  |   | 12 |    | 20 |    |
| 12 | 4 | 8  | 12 | 16 |    |
| 16 |   |    |    |    |    |

# Saugus kelio pasirinkimas



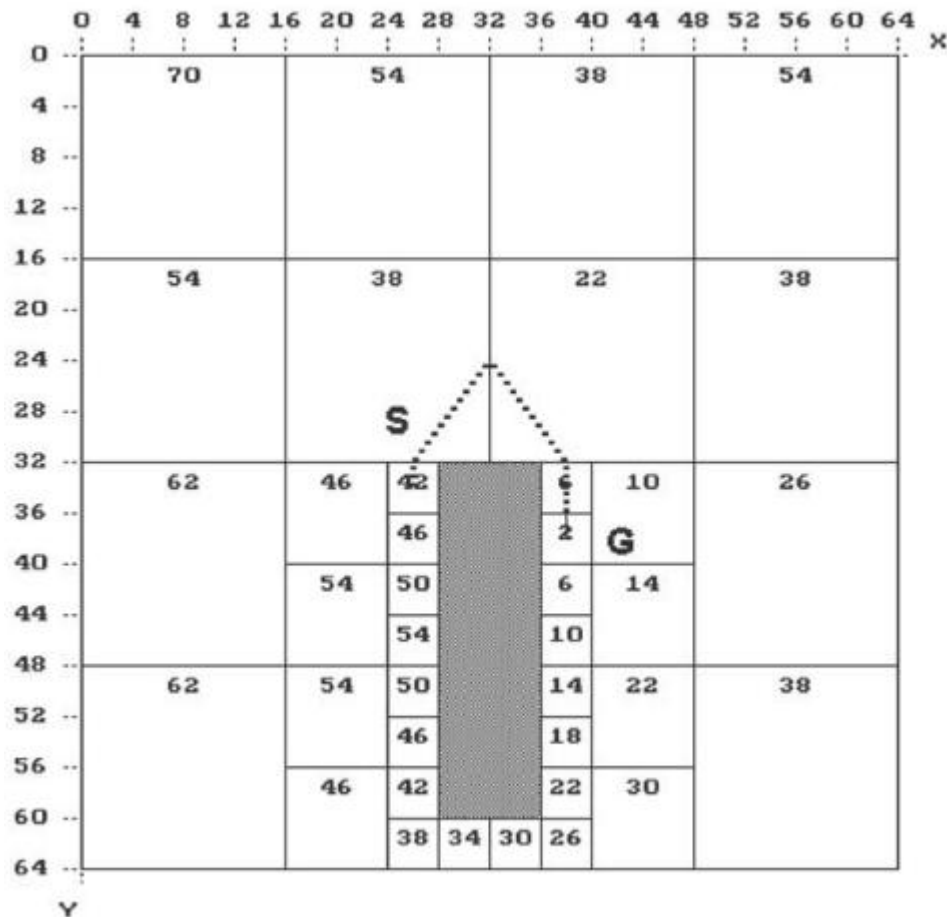
$$A = \begin{pmatrix} x_0 y_0 \\ x_1 y_1 \\ \vdots \\ x_n y_n \end{pmatrix}$$

$$x_A = \sum_{i=0}^n x_i \times 2^{N-i-1},$$

$$y_A = \sum_{i=0}^n y_i \times 2^{N-i-1},$$

$$d_A = 2^{N-n-1}.$$

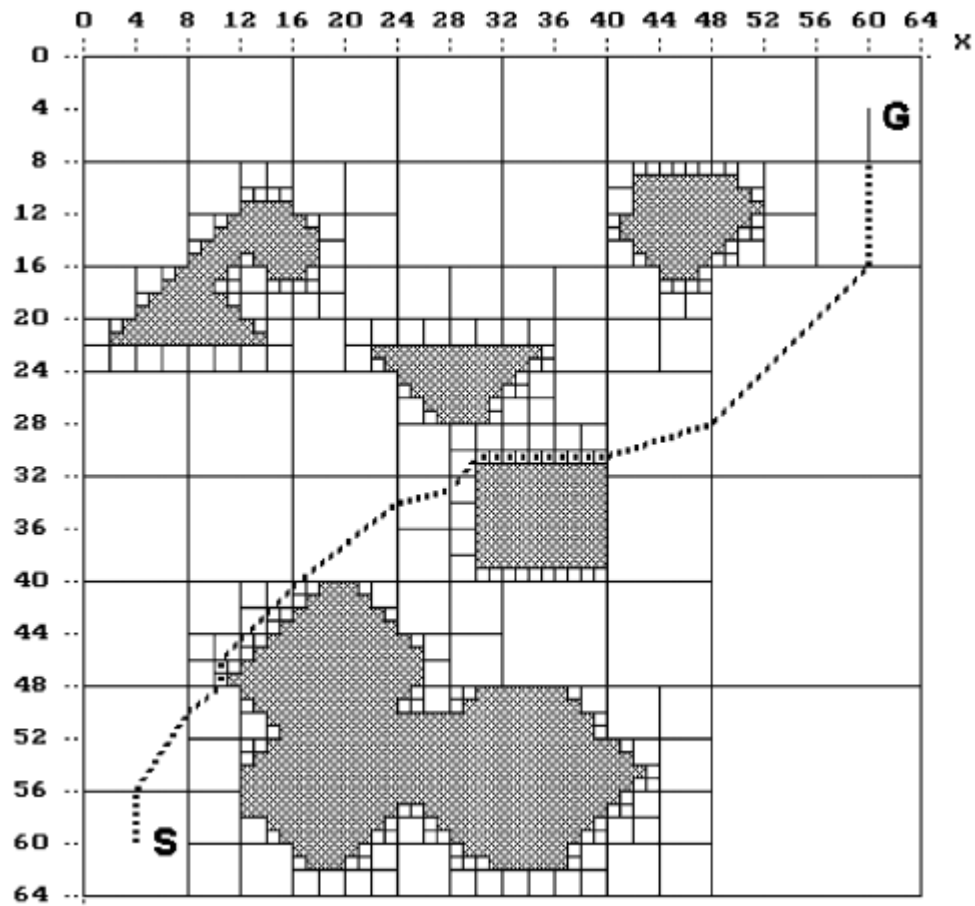
# Išplėstiniai ketvirtainiai medžiai



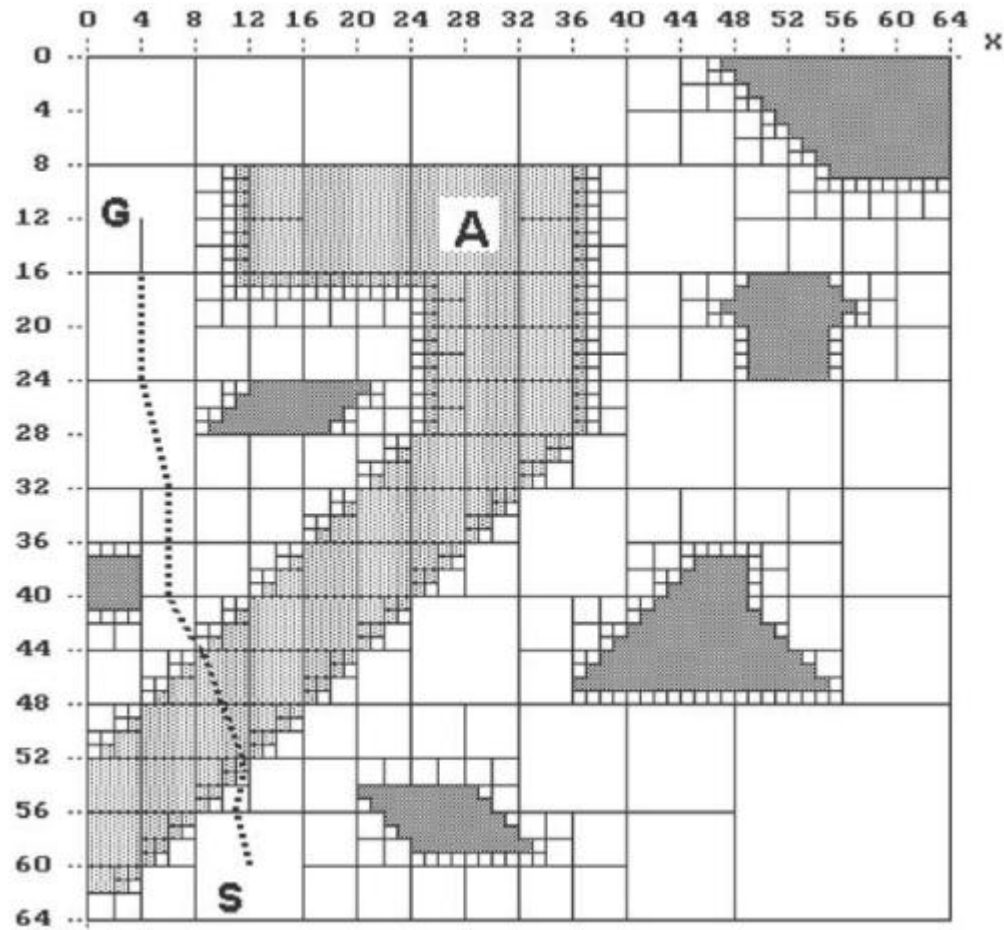
# Išplėstiniai ketvirtainiai medžiai

- Išplėstiniuose medžiuose pasirinkti blokai atvaizduojami pasirinktų medžio giliu.
- Pagrindinis trūkumas - medis užima daugiau atminties.
- Randamas optimalesnis kelias.

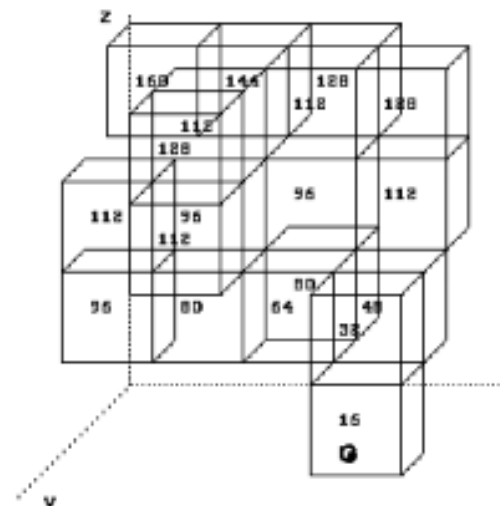
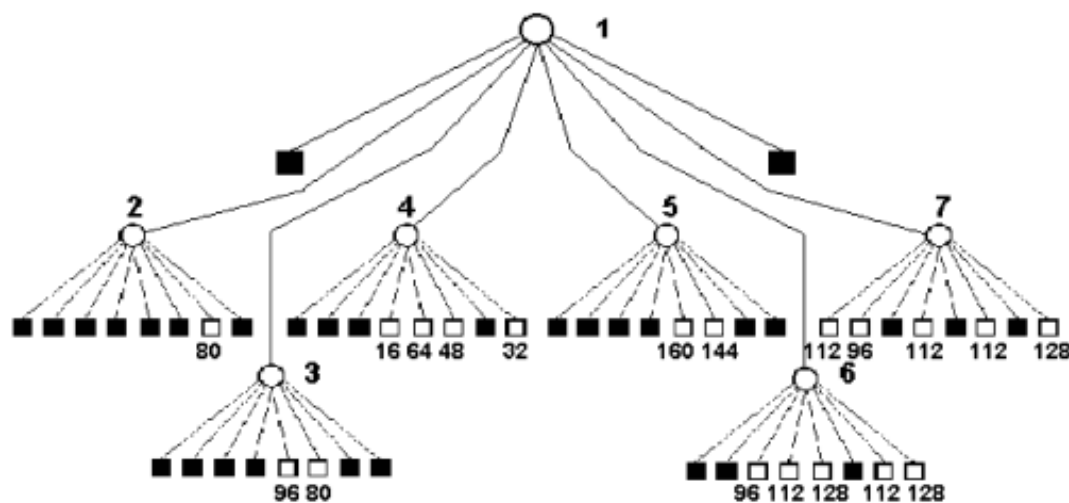
# Išplėstiniai ketvirtainiai medžiai



# Atstumų žemėlapis su svorio koeficientais

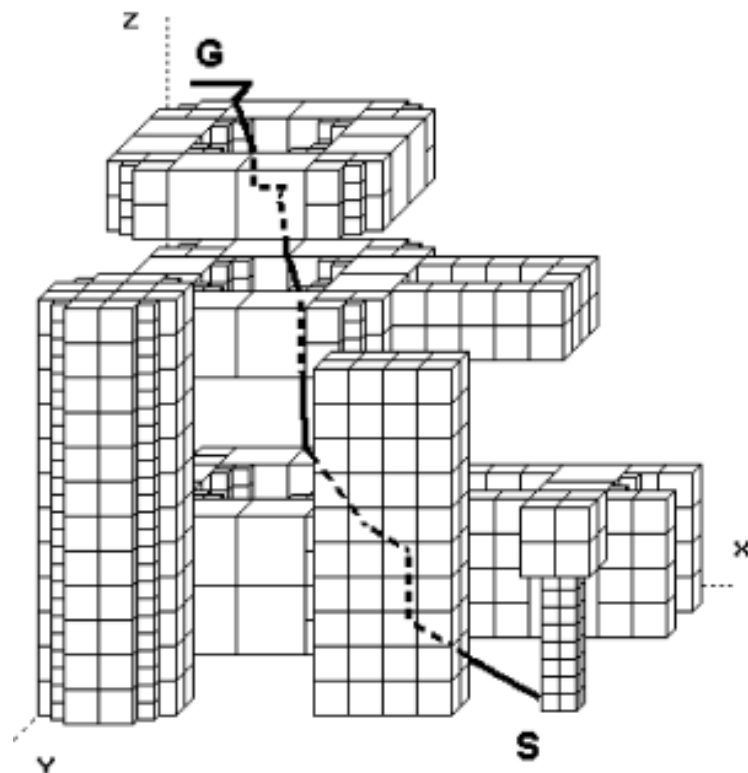


# Kelio planavimas erdvėje



| $F_i/S_{ij}$ | 0    | 1   | 2   | 3    | 4    | 5    | 6    | 7    |
|--------------|------|-----|-----|------|------|------|------|------|
| 1            | 0    | 2   | 3   | 4    | 5    | 6    | 7    | 0    |
| 2            | 0    | 0   | 0   | 0    | 0    | 0    | -80  | 0    |
| 3            | 0    | 0   | 0   | 0    | -96  | -80  | 0    | 0    |
| 4            | 0    | 0   | 0   | -16  | -64  | -48  | 0    | -32  |
| 5            | 0    | 0   | 0   | 0    | -160 | -144 | 0    | 0    |
| 6            | 0    | 0   | -96 | -112 | -128 | 0    | -112 | -128 |
| 7            | -112 | -96 | 0   | -112 | 0    | -112 | 0    | -128 |

# Kelio planavimas erdvėje





# Išvados

- Atstumų žemėlapius galima atvaizduoti pasinaudojant ketvirtainių ir aštuntainių medžių duomenų struktūras, naudojant žymiai mažiau atminties nei atvaizduojant rastinio failu.
- Šio metodo efektyvumas priklauso nuo kaimyninių blokų paieškos strategijos.

# Literatūra

Low-cost implementation of distance maps for path planning using matrix quadtrees and octrees. *Jozef Voros*.