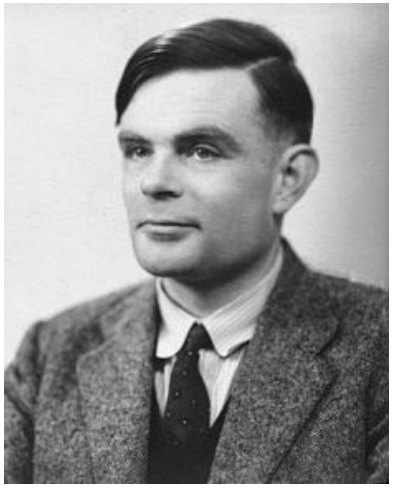
Rozwiązywanie zadań przez przeszukiwanie
A*, najszybszy wzrost, metoda Newtona

Jarosław Arabas

jaroslaw.arabas@pw.edu.pl
pok. 23

Artificial Intelligence

Dartmouth Summer Research Project on Artificial Intelligence 1956



acting reasonably	acting like humans
thinking reasonably	thinking like humans

(wg Russell, Norvig)

Słaba sztuczna inteligencja

Objaśnianie zjawisk

Przewidywanie zjawisk

Analiza scenariuszy

Uczenie maszynowe

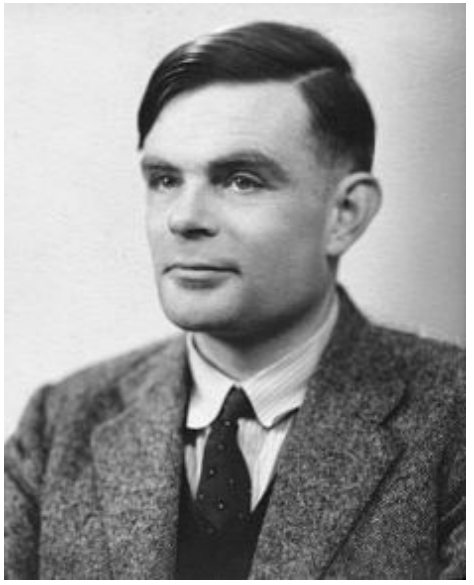
Tryb doradczy

Tryb autonomiczny

Podejmowanie decyzji

O co chodzi w AI?

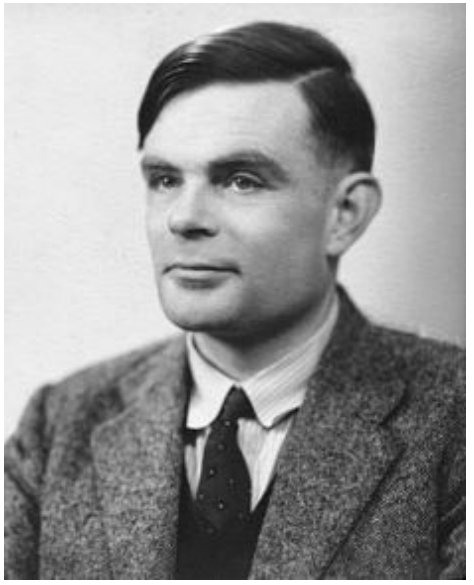
- uczenie z danych
- wspomaganie decyzji



- działanie nieodróżnialne od ludzkiego

O co chodzi w AI?

Dyskryminacyjna AI



Generatywna AI

Uczenie maszynowe

dane

```
1,273,82,105,210,9,904,680,23,62,34.99
2,163,149,191,180,12,843,746,0,20,41.14
3,162,148,191,179,16,840,743,1,20,41.81
4,162,148,190,179,19,838,741,3,21.5,42.08
5,154,112,144,220,10,923,658,20,64,26.82
6,147,89,115,202,9,860,829,23,55,25.21
7,152,139,178,168,18,944,695,0,20,38.86
8,145,0,227,240,6,750,853,14.5,58.5,36.59
9,152,0,237,204,6,785,892,15.5,51,32.71
10,304,0,140,214,6,895,722,19,51,38.46
11,145,106,136,208,10,751,883,24.5,61,26.02
12,148,109,139,193,7,768,902,23.75,58,28.03
13,142,130,167,215,6,735,836,25.5,67,31.37
14,354,0,0,234,6,959,691,17,54,33.91
15,374,0,0,190,7,1013,730,14.5,42.5,32.44
16,159,116,149,175,15,953,720,23.5,54.5,34.05
17,153,0,239,200,6,1002,684,12,35,28.29
18,295,106,136,206,11,750,766,25,68.5,41.01
19,310,0,143,168,10,914,804,20.5,48.2,49.3
20,296,97,0,219,9,932,685,15,48.5,29.23
21,305,100,0,196,10,959,705,20,49,29.77
22,310,0,143,218,10,787,804,13,46,36.19
23,148,180,0,183,11,972,757,0,20,18.52
24,146,178,0,192,11,961,749,18,46,17.19
25,142,130,167,174,11,883,785,0,20,36.72
26,140,128,164,183,12,871,775,23.75,53,33.38
27,308,111,142,217,10,783,686,25,70,42.08
28,295,106,136,208,6,871,650,26.5,70,39.4
29,298,107,137,201,6,878,655,16,26,41.27
30,314,0,161,207,6,851,757,21.5,64,41.14
31,321,0,164,190,5,870,774,24,60,45.82
32,349,0,178,230,6,785,721,20,68.5,43.95
33,366,0,187,191,7,824,757,24.75,62.7,52.65
34,274,89,115,202,9,759,827,26.5,68,35.52
35,137,167,214,226,6,708,757,27.5,70,34.45
36,275,99,127,184,13,810,790,25.75,64.5,43.54
37,252,76,97,194,8,835,821,23,54,33.11
38,165,150,0,182,12,1023,729,14.5,20,18.26
39,158,0,246,174,7,1035,706,19,43,34.99
40,156,0,243,180,11,1022,698,21,57,33.78
41,145,177,227,209,11,752,715,2.5,20,35.66
42,154,141,181,234,11,797,683,23,65,33.51
43,160,146,188,203,11,829,710,13,38,33.51
44,291,105,0,205,6,859,797,24,59,27.62
45,298,107,0,186,6,879,815,3,20,30.97
46,318,126,0,210,6,861,737,17.5,48,31.77
47,280,92,118,207,9,883,679,25.5,64,37.39
```

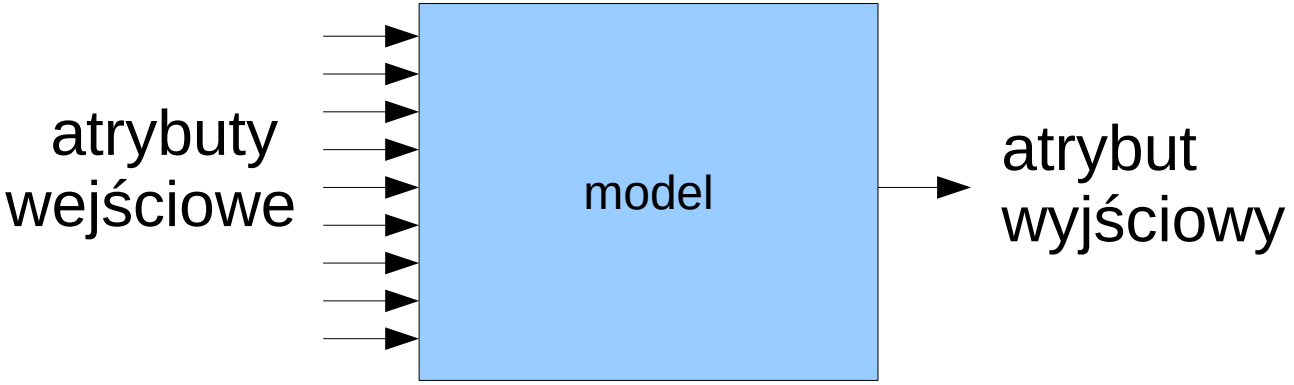
- Model ma agregować dane i przewidywać nieznane
- Milczące założenie:

zależności obecne w danych będą zachodzić w przyszłości

Uczenie maszynowe

dane

```
1,273,82,105,210,9,904,680,23,62,34.99
2,163,149,191,180,12,843,746,0,20,41.14
3,162,148,191,179,16,840,743,1,20,41.81
4,162,148,190,179,19,838,741,3,21.5,42.08
5,154,112,144,220,10,923,658,20,64,26.82
6,147,89,115,202,9,860,829,23,55,25.21
7,152,139,178,168,18,944,695,0,20,38.86
8,145,0,227,240,6,750,853,14.5,58.5,36.59
9,152,0,237,204,6,785,892,15.5,51,32.71
10,304,0,140,214,6,895,722,19,51,38.46
11,145,106,136,208,10,751,883,24.5,61,26.02
12,148,109,139,193,7,768,902,23.75,58,28.03
13,142,130,167,215,6,735,836,25.5,67,31.37
14,354,0,0,234,6,959,691,17,54,33.91
15,374,0,0,190,7,1013,730,14.5,42.5,32.44
16,159,116,149,175,15,953,720,23.5,54.5,34.05
17,153,0,239,200,6,1002,684,12,35,28.29
18,295,106,136,206,11,750,766,25,68.5,41.01
19,310,0,143,168,10,914,804,20.5,48.2,49.3
20,296,97,0,219,9,932,685,15,48.5,29.23
21,305,100,0,196,10,959,705,20,49,29.77
22,310,0,143,218,10,787,804,13,46,36.19
23,148,180,0,183,11,972,757,0,20,18.52
24,146,178,0,192,11,961,749,18,46,17.19
25,142,130,167,174,11,883,785,0,20,36.72
26,140,128,164,183,12,871,775,23.75,53,33.38
27,308,111,142,217,10,783,686,25,70,42.08
28,295,106,136,208,6,871,650,26.5,70,39.4
29,298,107,137,201,6,878,655,16,26,41.27
30,314,0,161,207,6,851,757,21.5,64,41.14
31,321,0,164,190,5,870,774,24,60,45.82
32,349,0,178,230,6,785,721,20,68.5,43.95
33,366,0,187,191,7,824,757,24.75,62.7,52.65
34,274,89,115,202,9,759,827,26.5,68,35.52
35,137,167,214,226,6,708,757,27.5,70,34.45
36,275,99,127,184,13,810,790,25.75,64.5,43.54
37,252,76,97,194,8,835,821,23,54,33.11
38,165,150,0,182,12,1023,729,14.5,20,18.26
39,158,0,246,174,7,1035,706,19,43,34.99
40,156,0,243,180,11,1022,698,21,57,33.78
41,145,177,227,209,11,752,715,2.5,20,35.66
42,154,141,181,234,11,797,683,23,65,33.51
43,160,146,188,203,11,829,710,13,38,33.51
44,291,105,0,205,6,859,797,24,59,27.62
45,298,107,0,186,6,879,815,3,20,30.97
46,318,126,0,210,6,861,737,17.5,48,31.77
47,280,92,118,207,9,883,679,25.5,64,37.39
```



Uczenie maszynowe

dane

```
1,273,82,105,210,9,904,680,23,62,34.99
2,163,149,191,180,12,843,746,0,20,41.14
3,162,148,191,179,16,840,743,1,20,41.81
4,162,148,190,179,19,838,741,3,21.5,42.08
5,154,112,144,220,10,923,658,20,64,26.82
6,147,89,115,202,9,860,829,23,55,25.21
7,152,139,178,168,18,944,695,0,20,38.86
8,145,0,227,240,6,750,853,14.5,58.5,36.59
9,152,0,237,204,6,785,892,15.5,51,32.71
10,304,0,140,214,6,895,722,19,51,38.46
11,145,106,136,208,10,751,883,24.5,61,26.02
12,148,109,139,193,7,768,902,23.75,58,28.03
13,142,130,167,215,6,735,836,25.5,67,31.37
14,354,0,0,234,6,959,691,17,54,33.91
15,374,0,0,190,7,1013,730,14.5,42.5,32.44
16,159,116,149,175,15,953,720,23.5,54.5,34.05
17,153,0,239,200,6,1002,684,12,35,28.29
18,295,106,136,206,11,750,766,25,68.5,41.01
19,310,0,143,168,10,914,804,20.5,48.2,49.3
20,296,97,0,219,9,932,685,15,48.5,29.23
21,305,100,0,196,10,959,705,20,49,29.77
22,310,0,143,218,10,787,804,13,46,36.19
23,148,180,0,183,11,972,757,0,20,18.52
24,146,178,0,192,11,961,749,18,46,17.19
25,142,130,167,174,11,883,785,0,20,36.72
26,140,128,164,183,12,871,775,23.75,53,33.38
27,308,111,142,217,10,783,686,25,70,42.08
28,295,106,136,208,6,871,650,26.5,70,39.4
29,298,107,137,201,6,878,655,16,26,41.27
30,314,0,161,207,6,851,757,21.5,64,41.14
31,321,0,164,190,5,870,774,24,60,45.82
32,349,0,178,230,6,785,721,20,68.5,43.95
33,366,0,187,191,7,824,757,24.75,62.7,52.65
34,274,89,115,202,9,759,827,26.5,68,35.52
35,137,167,214,226,6,708,757,27.5,70,34.45
36,275,99,127,184,13,810,790,25.75,64.5,43.54
37,252,76,97,194,8,835,821,23,54,33.11
38,165,150,0,182,12,1023,729,14.5,20,18.26
39,158,0,246,174,7,1035,706,19,43,34.99
40,156,0,243,180,11,1022,698,21,57,33.78
41,145,177,227,209,11,752,715,2.5,20,35.66
42,154,141,181,234,11,797,683,23,65,33.51
43,160,146,188,203,11,829,710,13,38,33.51
44,291,105,0,205,6,859,797,24,59,27.62
45,298,107,0,186,6,879,815,3,20,30.97
46,318,126,0,210,6,861,737,17.5,48,31.77
47,280,92,118,207,9,883,679,25.5,64,37.39
```

Dane są zasobem krytycznym

Uczenie maszynowe – czego dotyczy

Uczenie z danych o różnym poziomie “obiektywności”

Uczenie maszynowe – czego dotyczy

Uczenie z danych o różnym poziomie “obiektywności”

- Pomiarowych, opisujących świat “nieożywiony”

Uczenie maszynowe – czego dotyczy

Uczenie z danych o różnym poziomie “obiektywności”

- Pomiarowych, opisujących świat “nieożywiony”
- Pomiarowych, opisujacych interakcję człowieka ze światem

Uczenie maszynowe – czego dotyczy

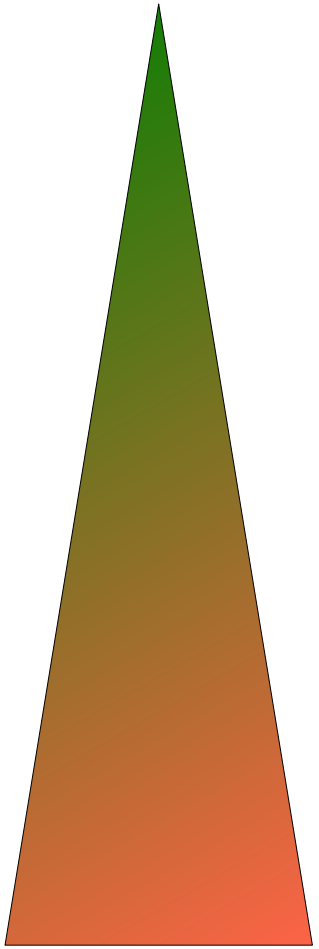
Uczenie z danych o różnym poziomie “obiektywności”

- Pomiarowych, opisujących świat “nieożywiony”

- Pomiarowych, opisujacych interakcję człowieka ze światem

- Wytworzonych przez człowieka tekstów i obrazów

Uczenie maszynowe – czego dotyczy



Uczenie z danych o różnym poziomie “obiektywności”

- Pomiarowych, opisujących świat “nieożywiony”
- Pomiarowych, opisujacych interakcję człowieka ze światem
- Wytworzonych przez człowieka tekstów i obrazów

Narastające trudności z weryfikacją sukcesu

Podejmowanie dezycji
Przykład: problem plecakowy

- n przedmiotów
- Każdy przedmiot ma wagę $w_i>0$ i wartość $p_i>0$
- Wybrać przedmioty o największej łącznej wartości i łącznej wadze nie większej niż W

$$max \sum_{i=1}^{n} x_i p_i$$

$$\sum_{i=1}^{n} x_i w_i \leqslant W$$

$$x_i \in \{0,1\}$$

Problem plecakowy

- n przedmiotów
- Każdy przedmiot ma wagę $w_i>0$ i wartość $p_i>0$
- Wybrać przedmioty o największej łącznej wartości i łącznej wadze nie większej niż W

$$max \sum_{i=1}^{n} x_i p_i$$

$$\sum_{i=1}^{n} x_i w_i \leqslant W$$

$$x_i \in \{0,1\}$$

Problem plecakowy – przykład

- Przedmioty

i	1	2	3	4
p_i	16	8	9	6
w_i	8	3	5	2
p_i/w_i	2	2.66	1.8	3

- W=9

$$max \sum_{i=1}^{n} x_i p_i$$

$$\sum_{i=1}^{n} x_i w_i \leqslant W$$

$$x_i \in \{0,1\}$$

Problem plecakowy – przykład

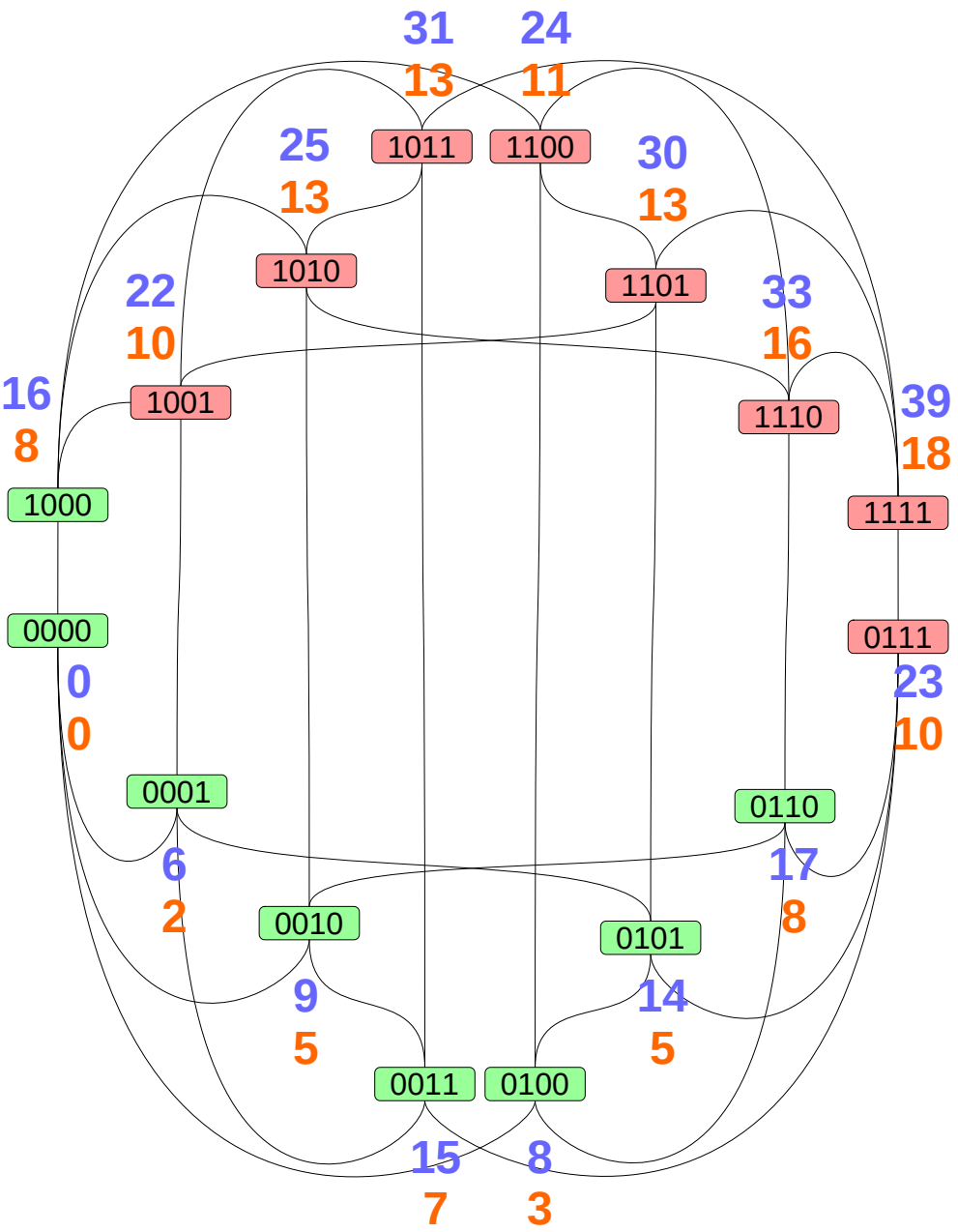
- Przedmioty

i	1	2	3	4
p_i	16	8	9	6
w_i	8	3	5	2
p_i/w_i	2	2.66	1.8	3

- W=9

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0	6	9	15	8	14	17	23	16	22	25	31	24	30	33	39
0	2	5	7	3	5	8	10	8	10	13	13	11	13	16	18

Przestrzeń przeszukiwań

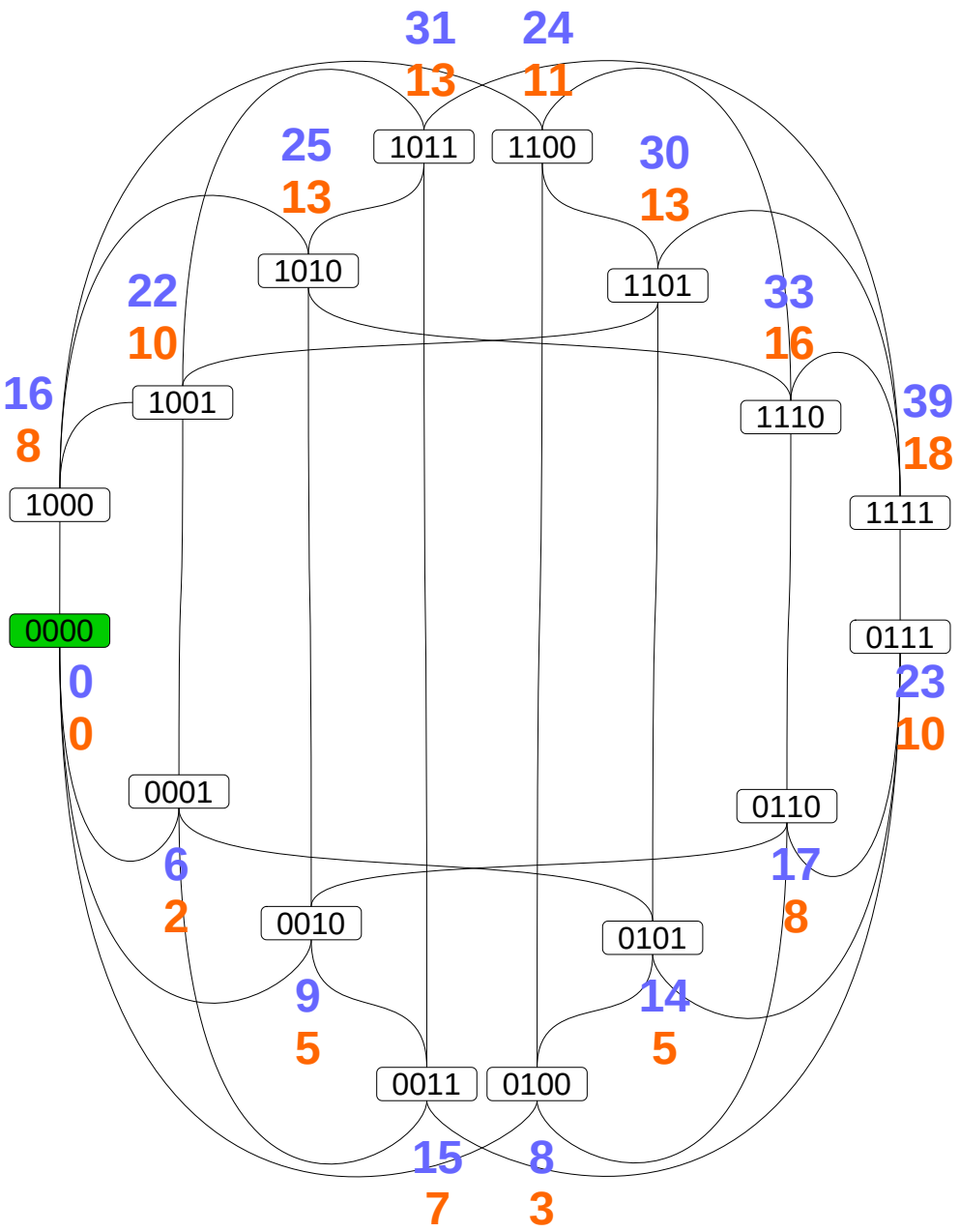


Algorytm wspinaczkowy

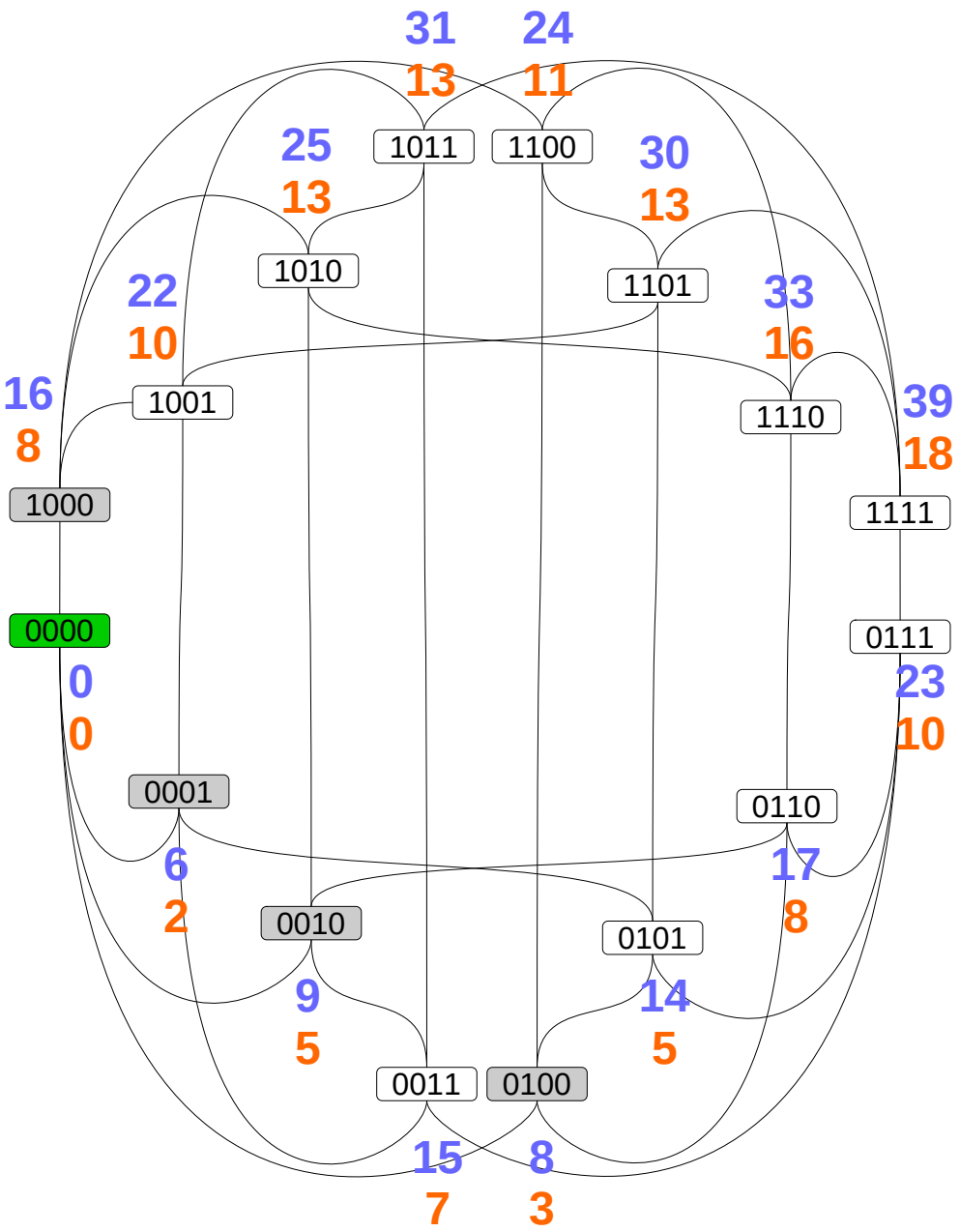
algorytm ***wspinaczkowy***
$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$
while *! stop*
  $Y \leftarrow N(\boldsymbol{x})$
  $\boldsymbol{y} \leftarrow wybierzNajlepszego(Y)$
  if $q(\boldsymbol{y}) > q(\boldsymbol{x})$
    $\boldsymbol{x} \leftarrow \boldsymbol{y}$

x – punkt roboczy

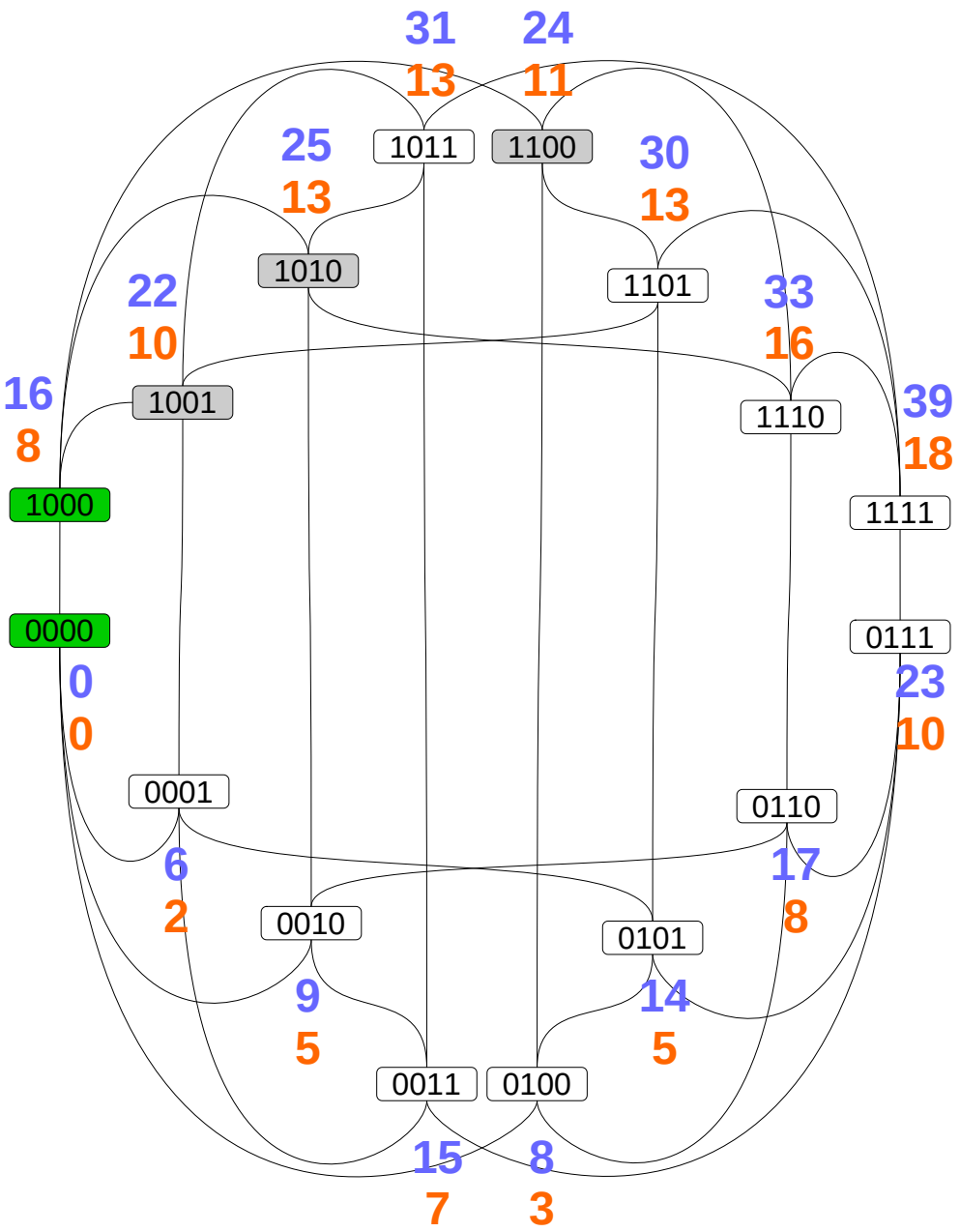
Algorytm wspinaczkowy



Algorytm wspinaczkowy



Algorytm wspinaczkowy



W R^n jest
continuum sąsiadów

Nie da się
wszystkich zbadać

Metoda najszybszego wzrostu *steepest ascent, gradient ascent*

algorytm ***największego wzrostu***

$\boldsymbol{x} \leftarrow \boldsymbol{x_0}$

while *! stop*

$\boldsymbol{d} \leftarrow \nabla q(\boldsymbol{x})$

$\boldsymbol{x} \leftarrow \boldsymbol{x} + \beta_t \boldsymbol{d}$

x – punkt roboczy

$\nabla q(\boldsymbol{x})$ Gradient funkcji celu w punkcie x

β_t Ciąg dodatnich współczynników kroku

$$\nabla q(\boldsymbol{x}) = \begin{bmatrix} \partial q(\boldsymbol{x})/\partial x_1 \\ \vdots \\ \partial q(\boldsymbol{x})/\partial x_i \\ \vdots \\ \partial q(\boldsymbol{x})/\partial x_n \end{bmatrix}$$

Metoda najszybszego wzrostu *steepest ascent, gradient ascent*

algorytm ***największego wzrostu***

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$

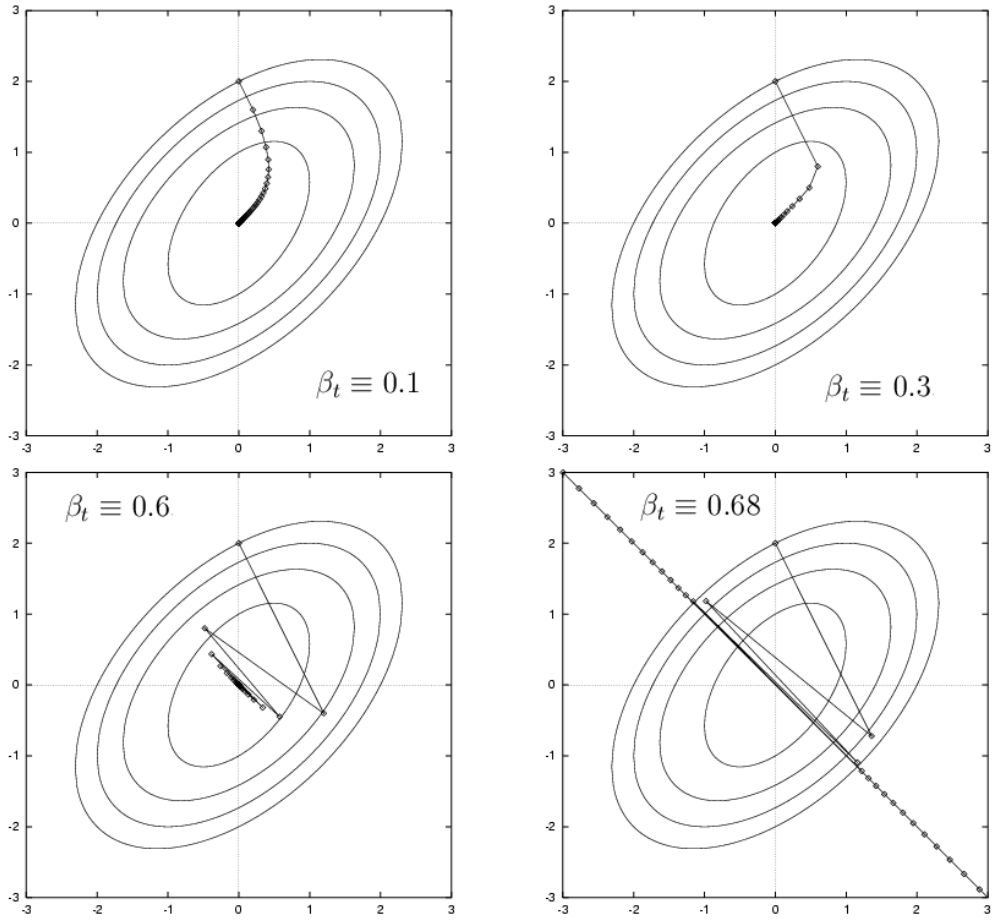
while *! stop*

$\boldsymbol{d} \leftarrow \nabla q(\boldsymbol{x})$

$\boldsymbol{x} \leftarrow \boldsymbol{x} + \beta_t \boldsymbol{d}$

W praktyce gradient jest estymowany np. przez losowe próbkowanie

Wówczas mówimy o stochastycznym najszybszym wzroście/spadku



Metoda Newtona

algorytm ***metoda Newtona***

$\boldsymbol{x} \leftarrow \boldsymbol{x_0}$

while *! stop*

$\boldsymbol{d} \leftarrow H_q^{-1}(\boldsymbol{x}) \cdot \nabla q(\boldsymbol{x})$

$\boldsymbol{x} \leftarrow \boldsymbol{x} + \beta \boldsymbol{d}$

x – punkt roboczy

$H_q(\boldsymbol{x})$ hesjan funkcji celu w punkcie x

$$H_q(\boldsymbol{x}) = \begin{bmatrix} \partial^2 q(\boldsymbol{x}) / (\partial x_1)^2 & \partial^2 q(\boldsymbol{x}) / (\partial x_1 \cdot \partial x_2) & \cdots \\ \partial^2 q(\boldsymbol{x}) / (\partial x_1 \cdot \partial x_2) & \partial^2 q(\boldsymbol{x}) / (\partial x_2)^2 & \cdots \\ \vdots & \vdots & \ddots \end{bmatrix}$$

W praktyce hesjan jest aproksymowany – metody BFGS, DFP zwane metodami zmiennej metryki

Metoda Newtona

algorytm ***największego wzrostu***

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$

while *! stop*

$\boldsymbol{d} \leftarrow \nabla q(\boldsymbol{x})$

$\boldsymbol{x} \leftarrow \boldsymbol{x} + \beta \boldsymbol{d}$

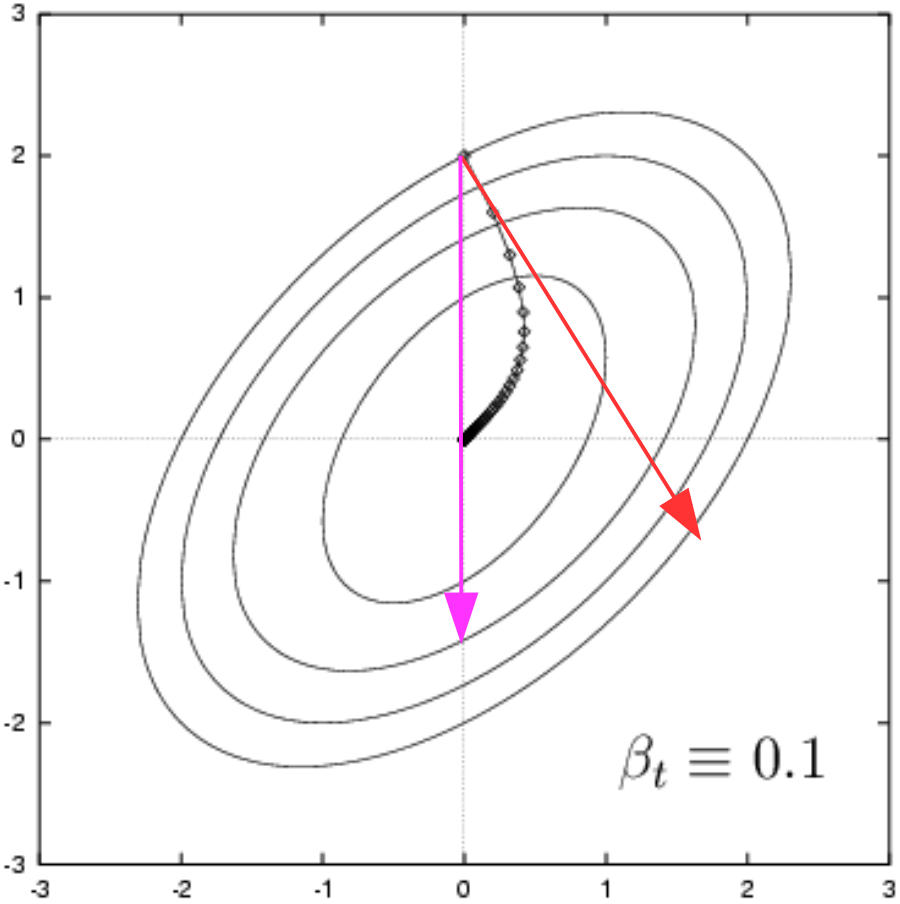
algorytm ***metoda Newtona***

$\boldsymbol{x} \leftarrow \boldsymbol{x}_0$

while *! stop*

$\boldsymbol{d} \leftarrow H_q^{-1}(\boldsymbol{x}) \cdot \nabla q(\boldsymbol{x})$

$\boldsymbol{x} \leftarrow \boldsymbol{x} + \beta \boldsymbol{d}$



Problem plecakowy

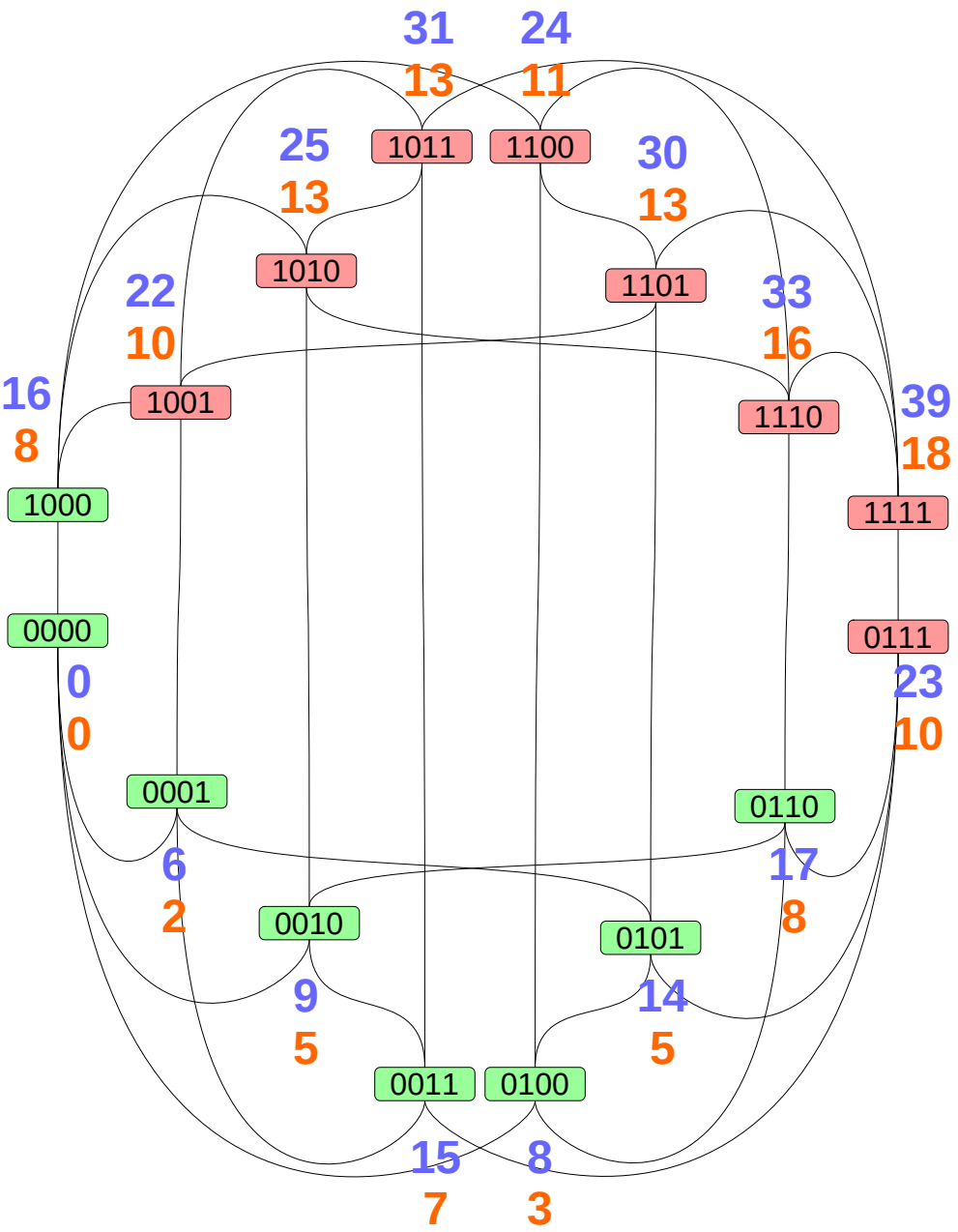
- n przedmiotów
- Każdy przedmiot ma wagę $w_i>0$ i wartość $p_i>0$
- Wybrać przedmioty o największej łącznej wartości i łącznej wadze nie większej niż W

$$max \sum_{i=1}^{n} x_i p_i$$

$$\sum_{i=1}^{n} x_i w_i \leqslant W$$

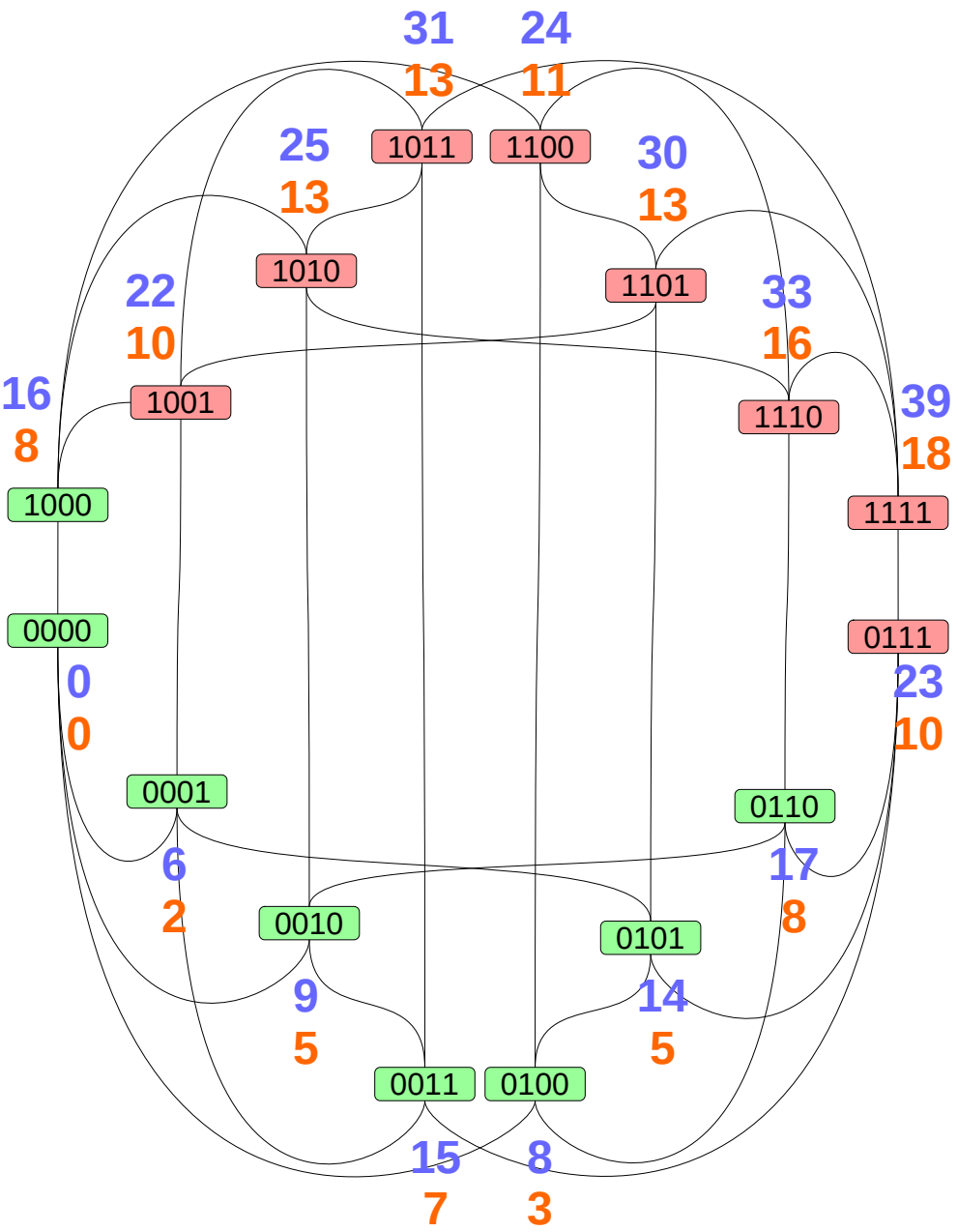
$$x_i \in \{0,1\}$$

Przestrzeń przeszukiwań

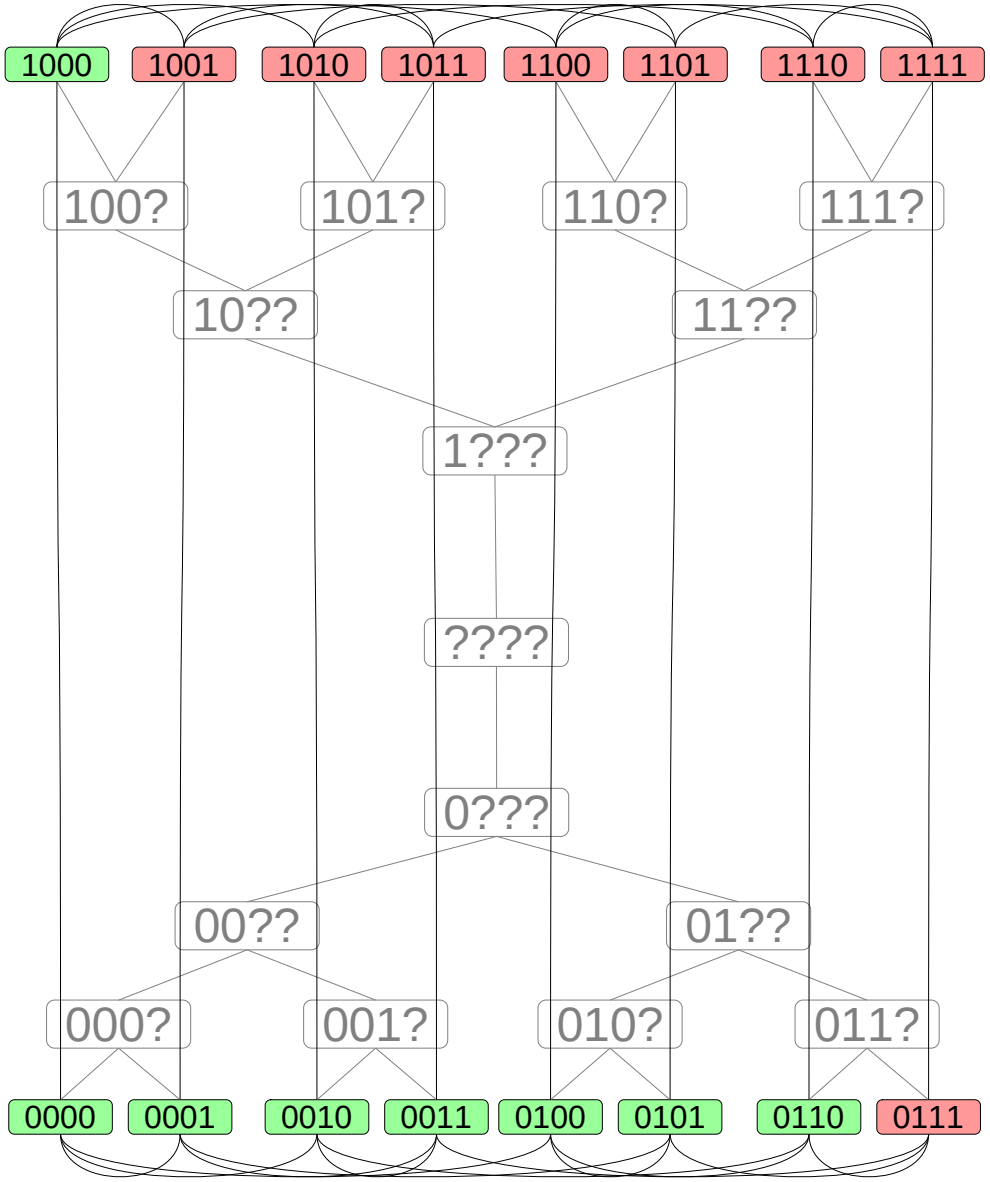


Popatrzmy
z innej
perspektywy...

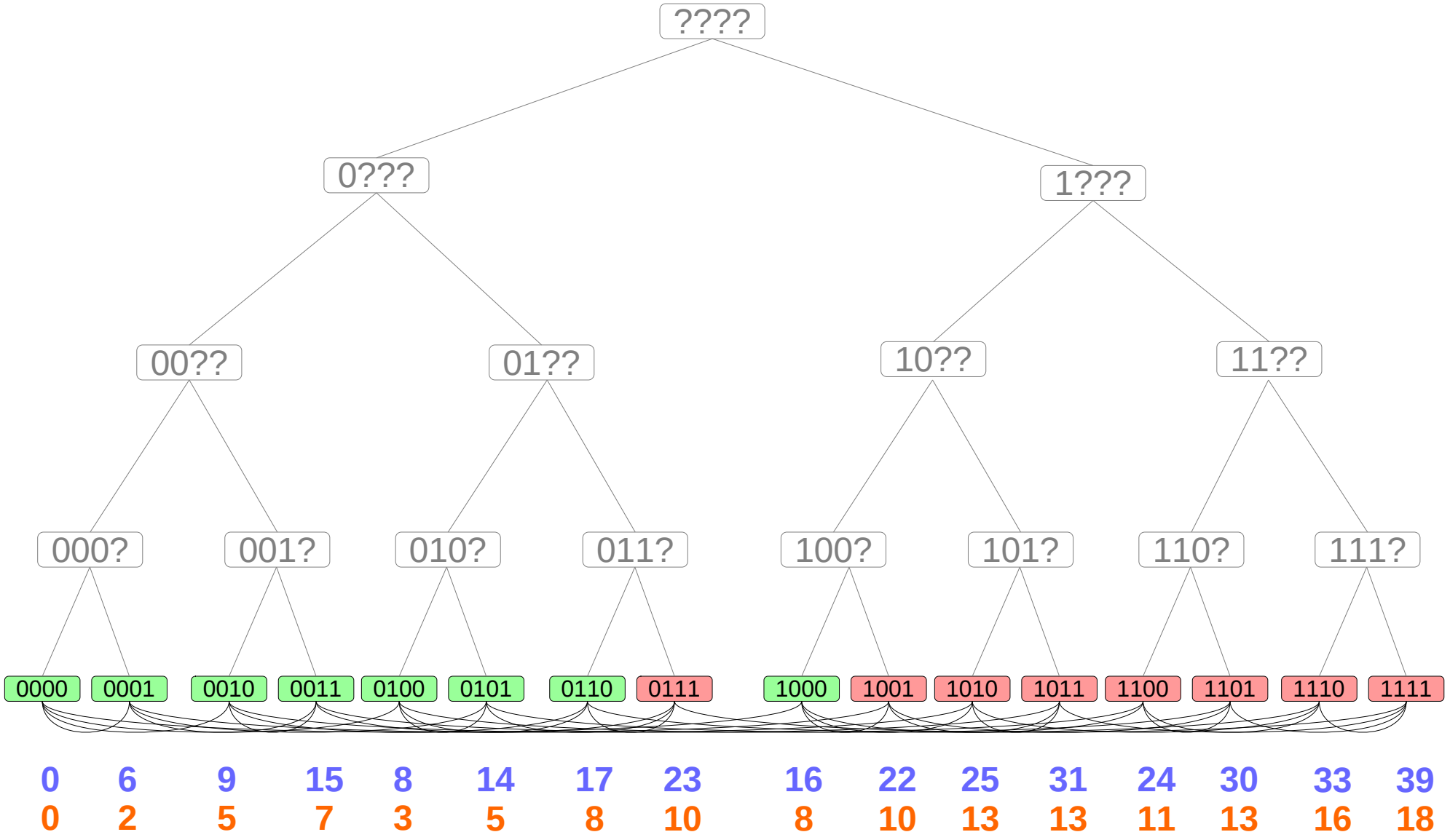
Przestrzeń przeszukiwań



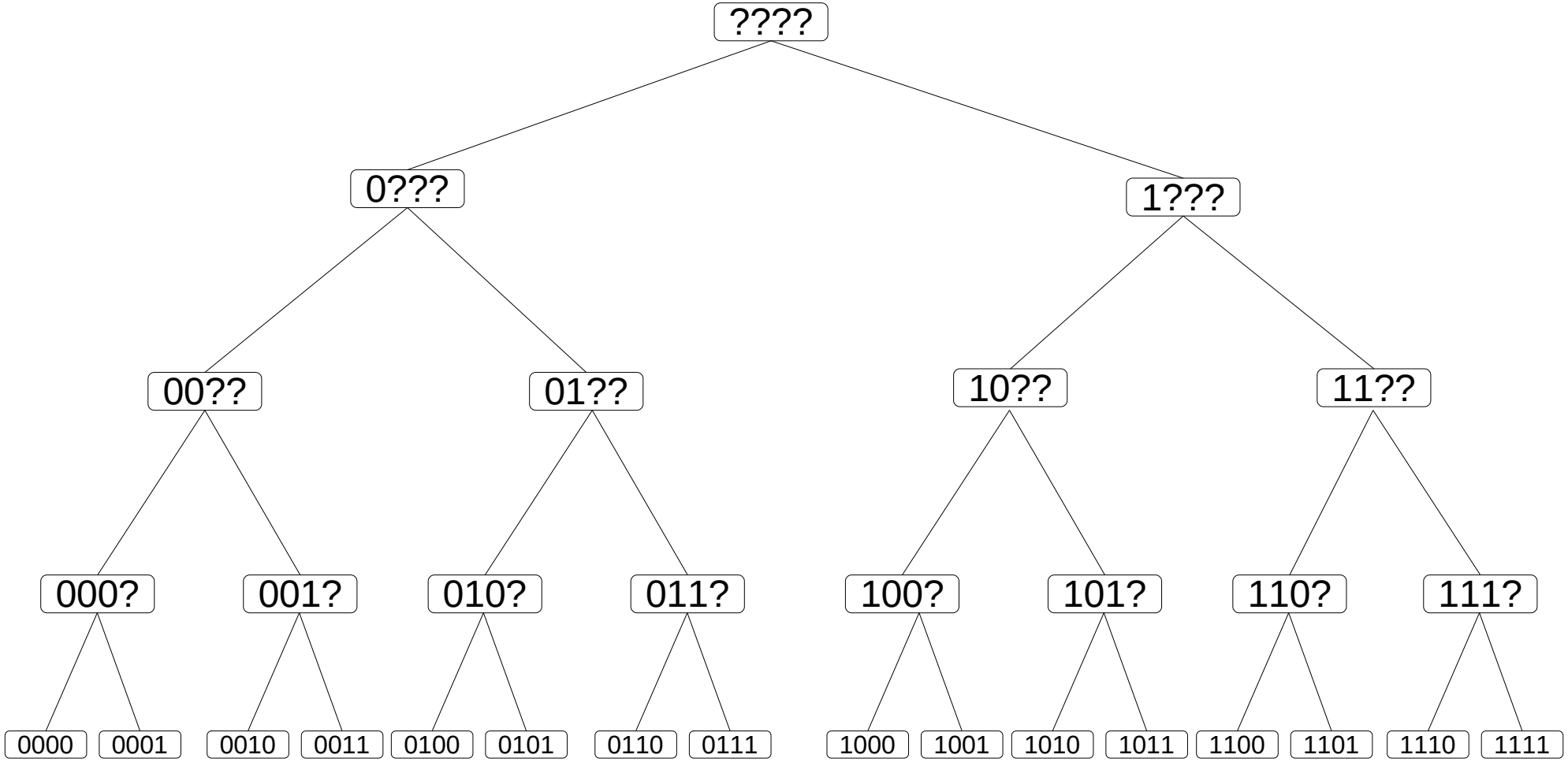
Przestrzeń przeszukiwań



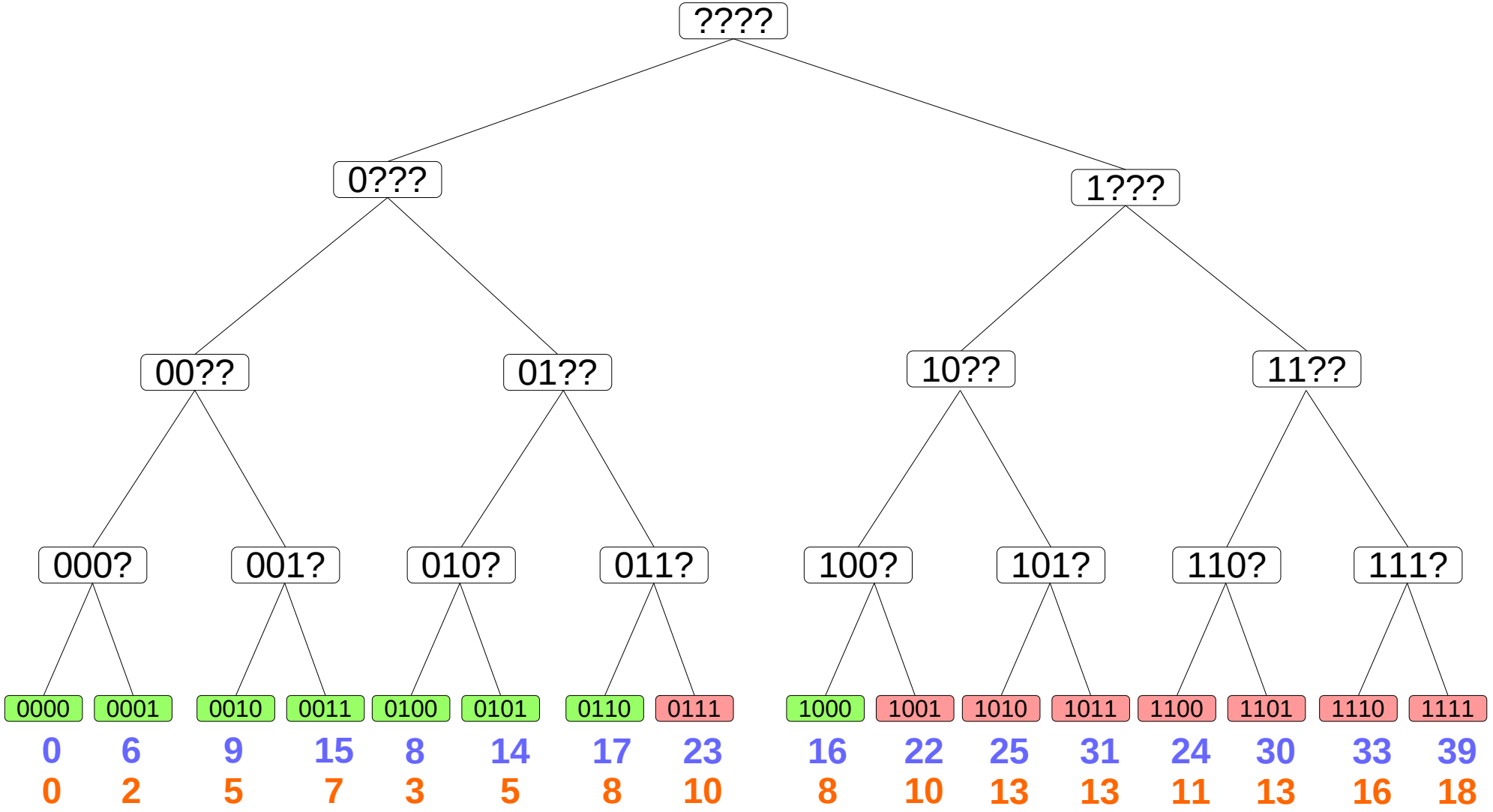
Przestrzeń przeszukiwań



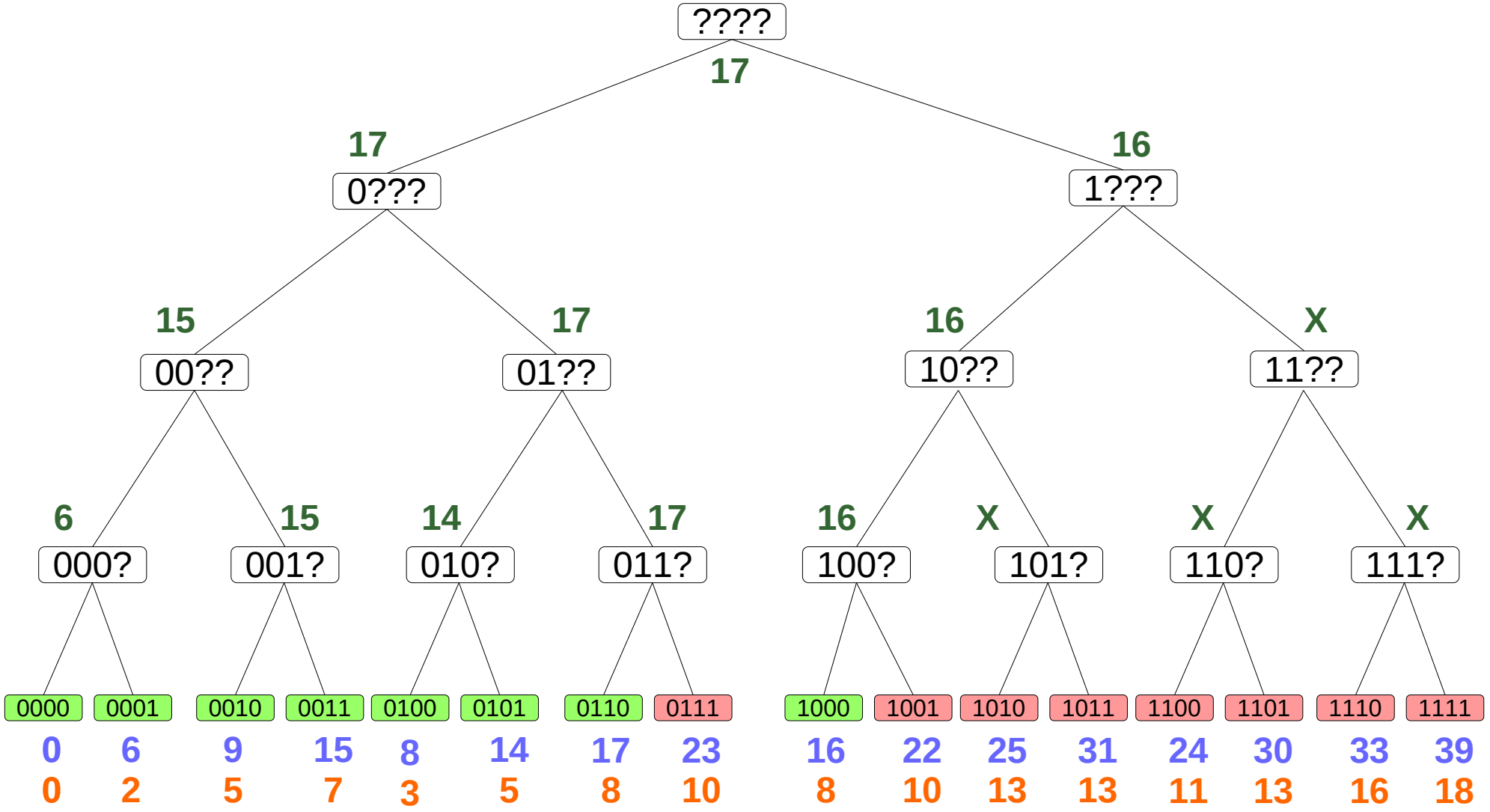
Przestrzeń przeszukiwań



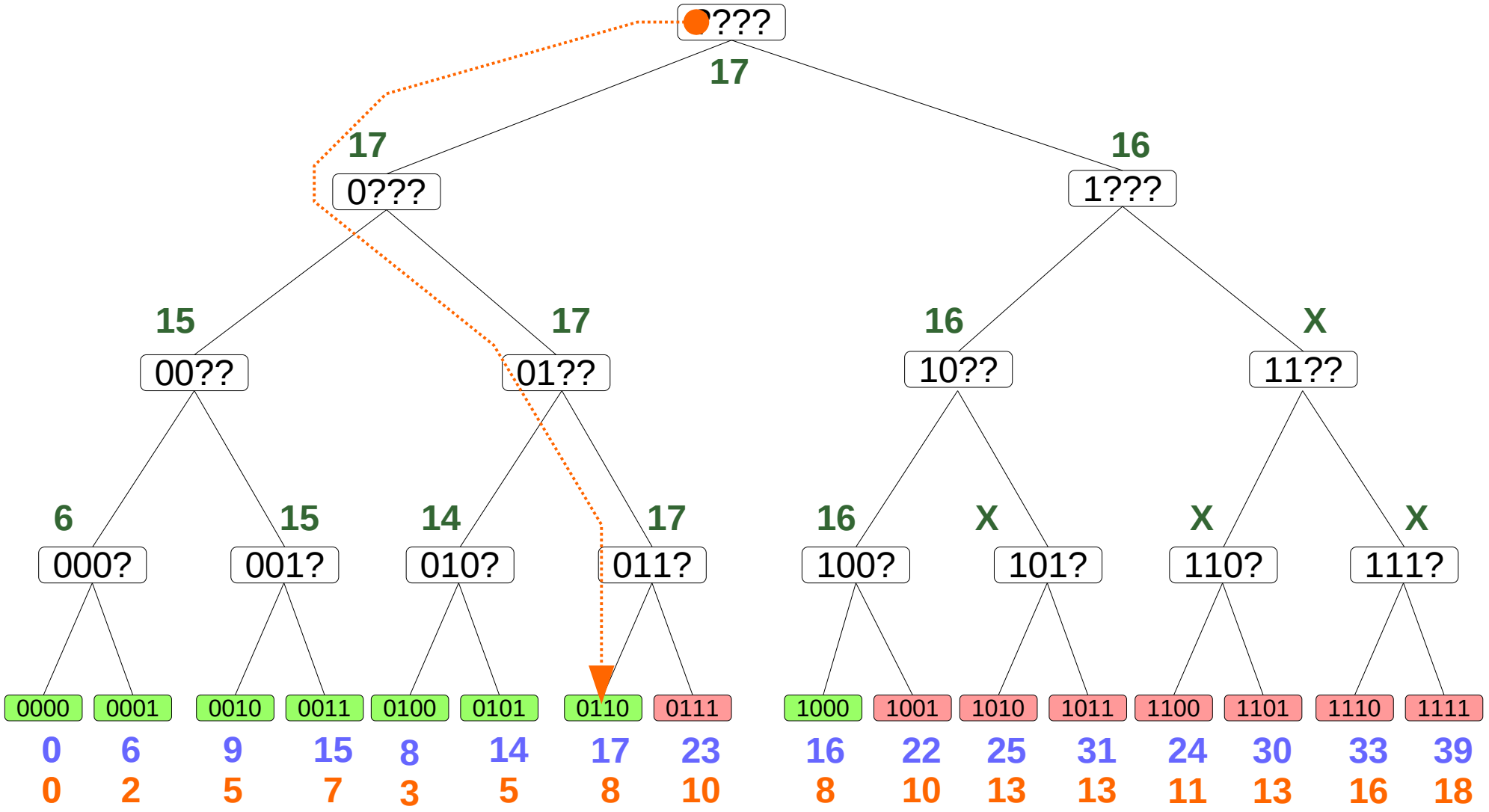
Funkcja celu, zbiór dopuszczalny



Propagacja wartości f.celu “od tyłu”



Rady Dobrej Wróżki

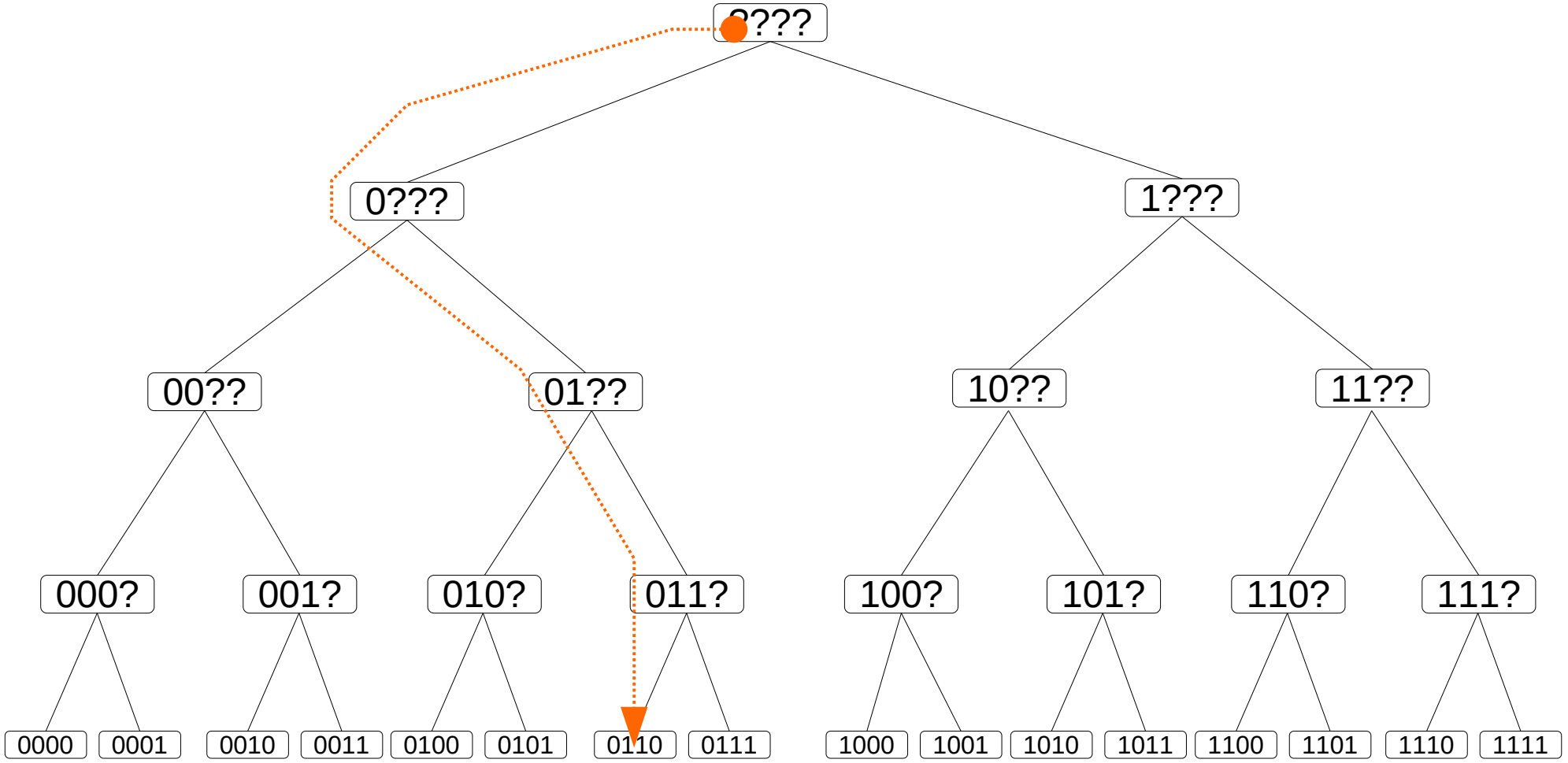


Algorytm najpierw najlepszy

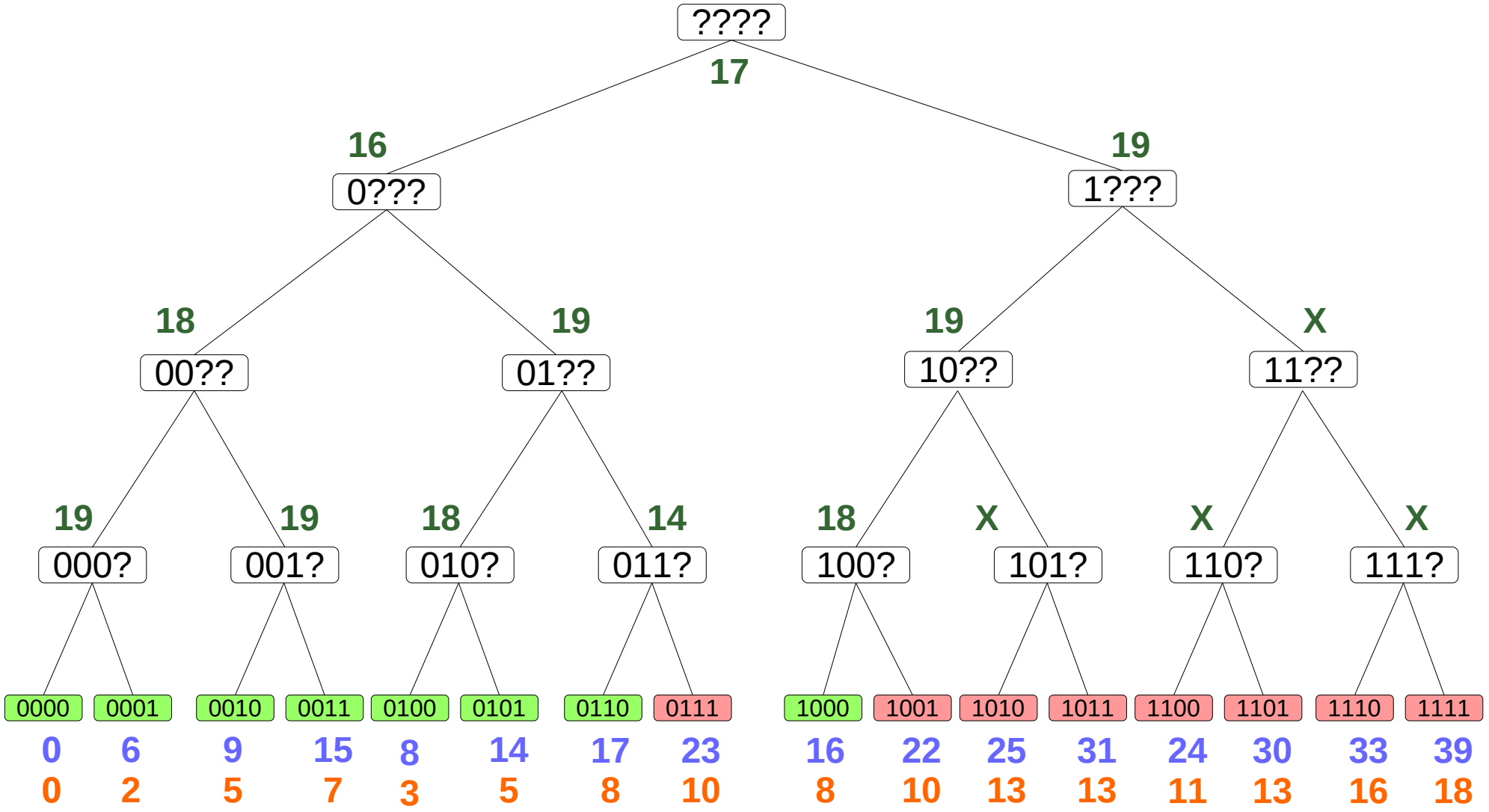
algorytm ***najpierw najlepszy***
$A \leftarrow init(s_0)$
while $!\, stop$
  $\boldsymbol{x} \leftarrow popPriorityQueue(A)$
  $Y \leftarrow dzieci(\boldsymbol{x})$
  $A \leftarrow A \cup Y$

- Jeśli etykiety są dokładne, to możemy algorytm zatrzymać po dotarciu do pierwszego terminalnego węzła

Jak uzyskać rady Dobrej Wróżki?



Rady niedobrej wróżki



Algorytm najpierw najlepszy

algorytm ***najpierw najlepszy***
$A \leftarrow init(s_0)$
while $!\, stop$
  $\boldsymbol{x} \leftarrow popPriorityQueue(A)$
  $Y \leftarrow dzieci(\boldsymbol{x})$
  $A \leftarrow A \cup Y$

- Jeśli etykiety są niedokładne, to algorytm musi wyczerpać listę A aby dać gwarancję uzyskania wyniku optymalnego

WANTED

an oracle

Funkcja heurystyczna (dla problemu maksymalizacji)

- Dla problemu plecakowego, ostateczny zysk jest sumą zysków dających się wyznaczyć niezależnie dla każdej decycji

$$g(\boldsymbol{x}) = g_1(x_1) + g_2(x_2) + \ldots + g_n(x_n)$$

- Można dokonać oszacowania dla sekwencji k podjętych decycji (‘?’ oznacza niepodjętą decyzję):

$$g(x_1 \ldots x_k, ?_{k+1} \ldots ?_n) \geq \sum_{i=1\ldots k} g_i(x_i)$$

$$g(x_1 \ldots x_k, ?_{k+1} \ldots ?_n) \leq \sum_{i=1\ldots k} g_i(x_i) + \sum_{i=k+1\ldots n} \max_{y_n} g_i(y_i)$$

To górne ograniczenie jest niedokładne, spróbujmy je poprawić

Funkcja heurystyczna

- Idealne górne oszacowanie przyrostu zysku dla niepodjętych decyzji (F jest zbiorem dopuszczalnych liści)

$$g(x_1 \dots x_k, ?_{k+1} \dots ?_n) \leq \sum_{i=1\dots k} g_i(x_i) + \max_{\mathbf{y}: x_1 \dots x_k, y_{k+1} \dots y_n \in F} \sum_{i=k+1\dots n} g_i(y_i)$$

- Funkcja heurystycza h jest oszacowaniem nieidealnym:

$$g(x_1 \dots x_k, ?_{k+1} \dots ?_n) \leq \sum_{i=1\dots k} g_i(x_i) + h(x_1 \dots x_k, ?_{k+1} \dots ?_n)$$

  - Powinna być dopuszczalna

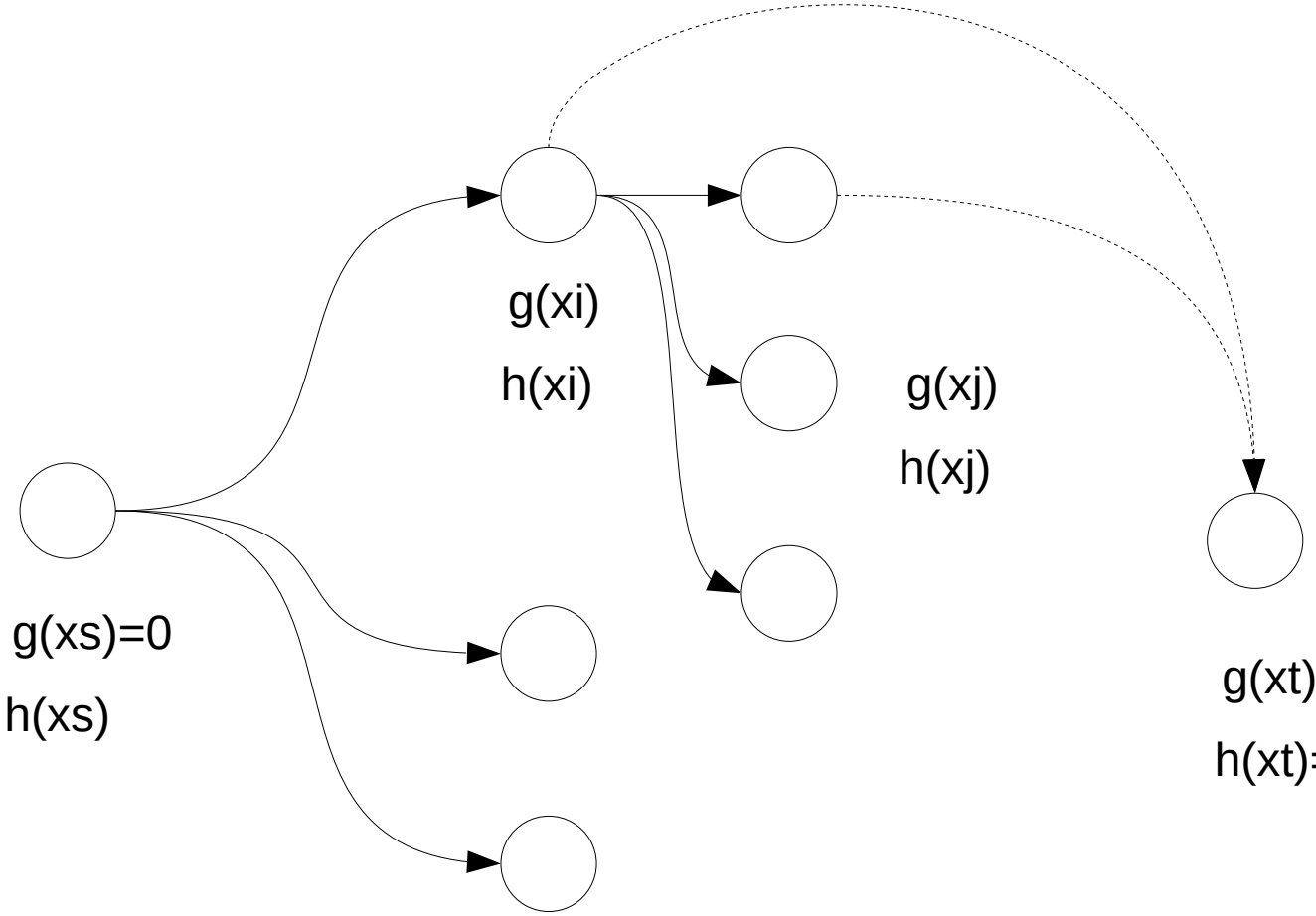
$$h(x_1 \dots x_k, ?_{k+1} \dots ?_n) \geq \max_{\mathbf{y}: x_1 \dots x_k, y_{k+1} \dots y_n \in F} \sum_{i=k+1\dots n} g_i(y_i)$$

  - Powinna być monotoniczna

$$\sum_{i=1\dots k+1} g_i(x_i) + h(x_1 \dots x_{k+1}, ?_{k+2} \dots ?_n) \leq \sum_{i=1\dots k+1} g_i(x_i) + h(x_1 \dots x_k, ?_{k+1} \dots ?_n)$$

Kierunki nierówności są przyjęte dla zadania maksymalizacji, przy minimalizacji przyjąć przeciwne

Funkcja heurystyczna (dla problemu maksymalizacji)



Nadmierny optymizm

Błąd oszacowania malejący wraz ze zbliżaniem się do rozwiązania

dopuszczalność: $g(x)+h(x)>=g(xt)$

monotoniczność: $g(xj)+h(xj)<=g(xi)+h(xi)$

Algorytm A*

Algorytm najpierw najlepszy
+
Funkcja kosztu
+
Funkcja heurystyczna
=
Algorytm A*

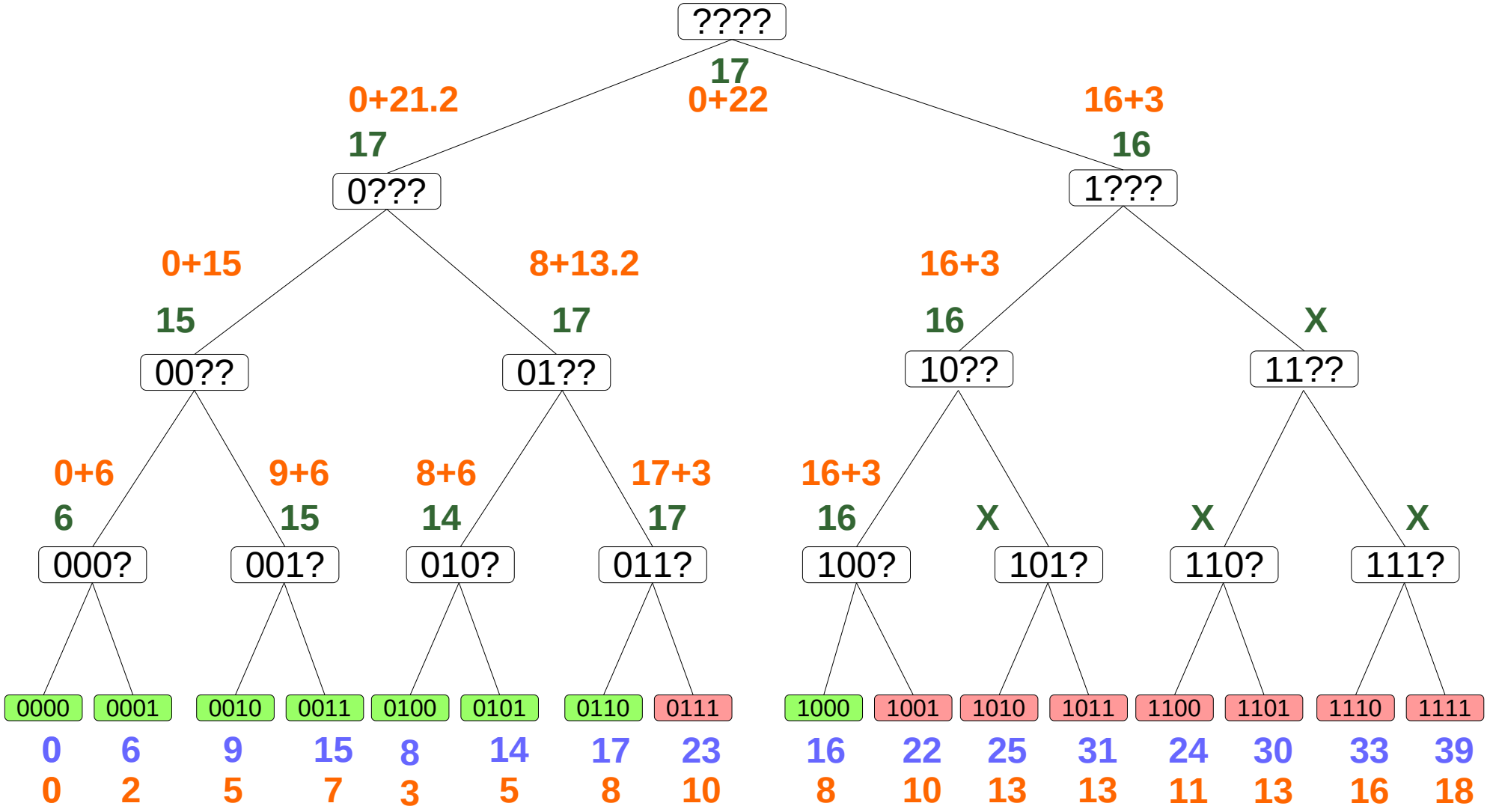
Problem plecakowy funkcja heurystyczna

- Przedmioty

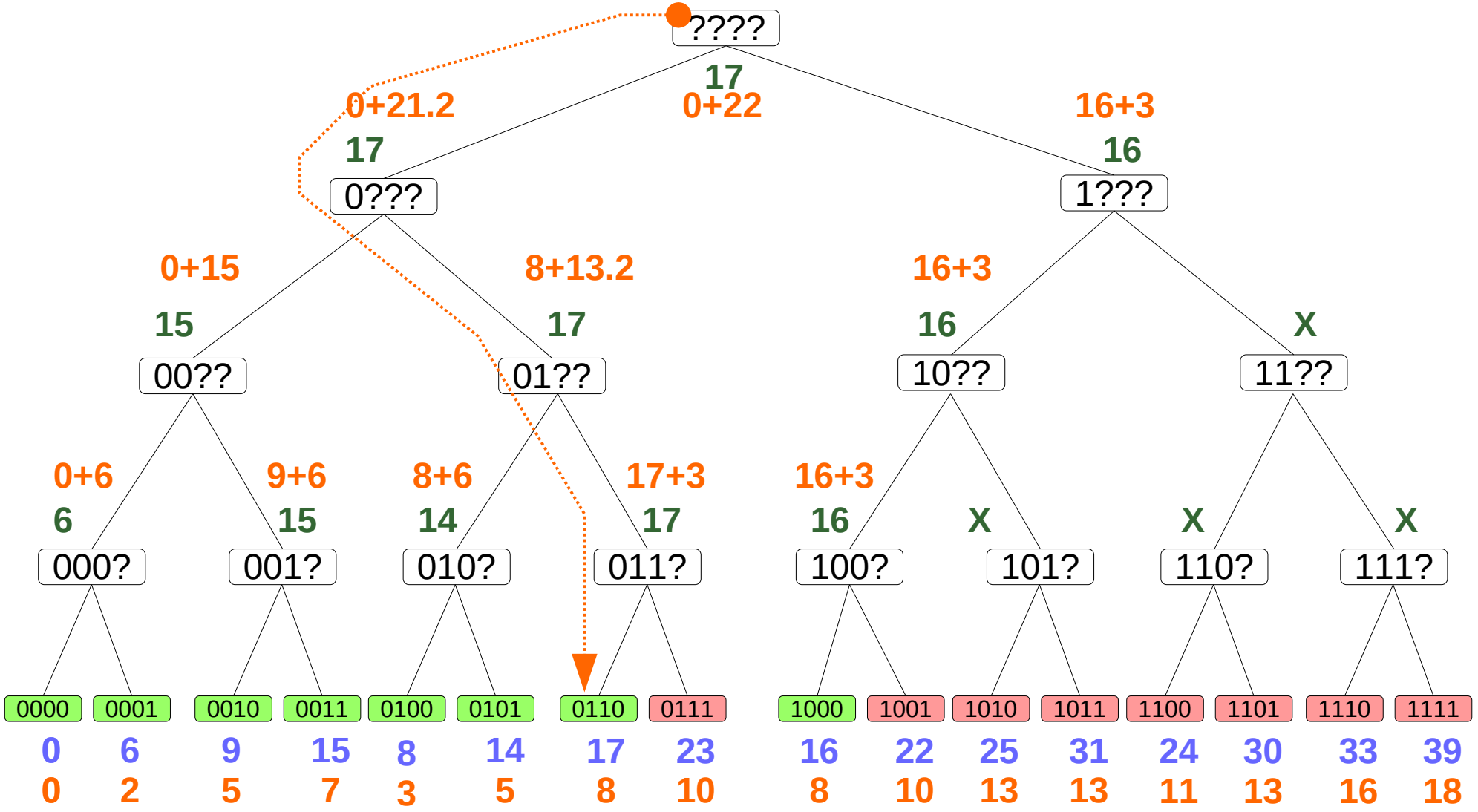
i	1	2	3	4
p_i	6	8	9	16
w_i	2	3	5	8
p_i/w_i	3	2.66	1.8	2

- W=9
- Przykładowe rozwiązanie $x = 1???$
- Łączna wartość: $g(x) = p_1 = 6$
- Łączna waga: $w(x) = w_1 = 2$
- Podziały przedmiotów: $y = [0,1,0,1/2]$
- Funkcja heurystyczna: $h(x) = 16 \cdot 1/2 + 8 \cdot 1 = 16$

Problem plecakowy
funkcja zysku i funkcja heurystyczna



Problem plecakowy
funkcja zysku i funkcja heurystyczna



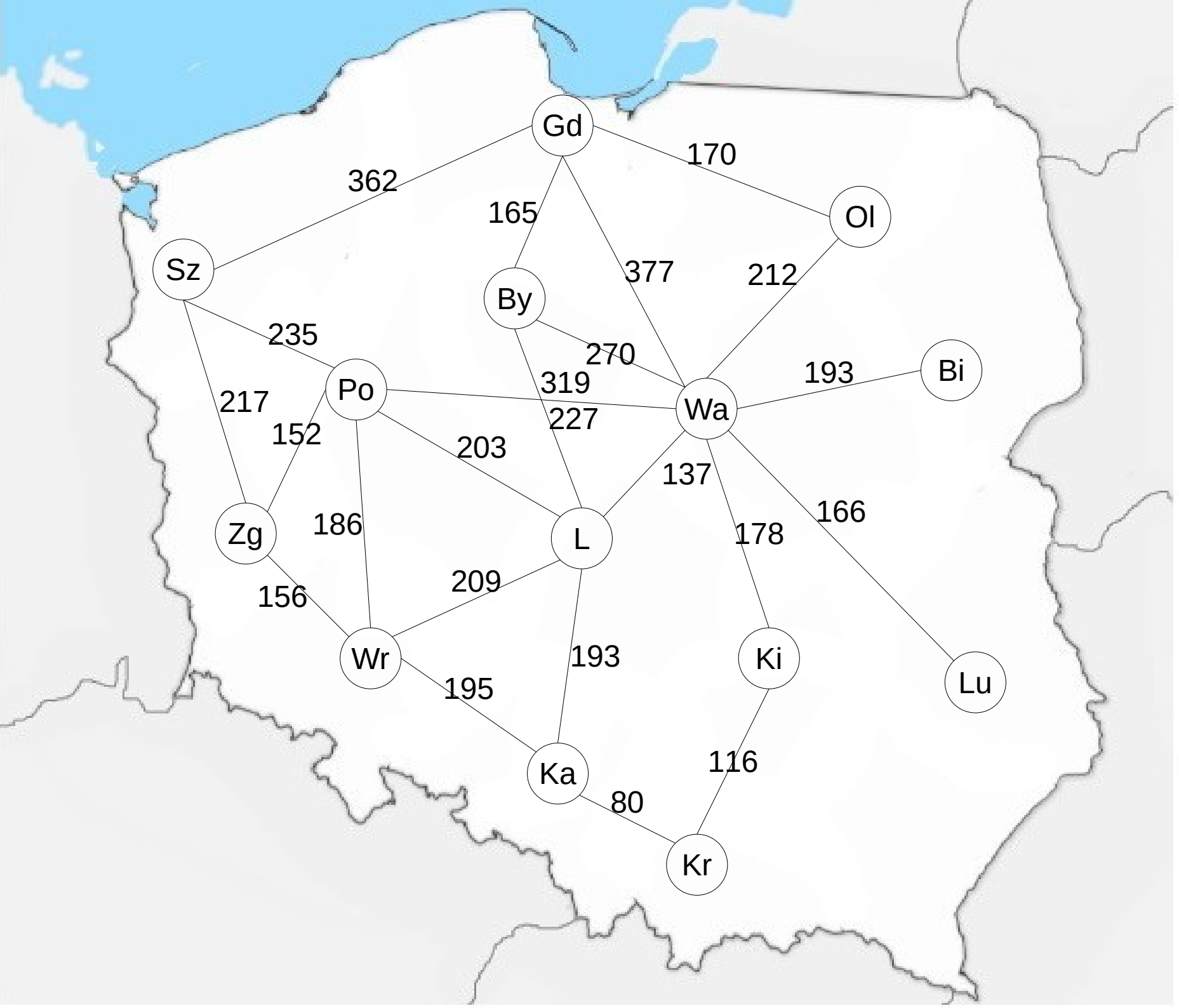
Algorytm najpierw najlepszy

algorytm ***najpierw najlepszy***
$A \leftarrow init(s_0)$
$best \leftarrow 0$
while $!\, stop$
  $\boldsymbol{x} \leftarrow popPriorityQueue(A)$
  if $terminal(x) \wedge (g(x) > best)$
    $best \leftarrow g(x)$
  if $!\, terminal(x) \wedge (g(x) + h(x) \geq best)$
    $Y \leftarrow dzieci(\boldsymbol{x})$
    $A \leftarrow A \cup Y$

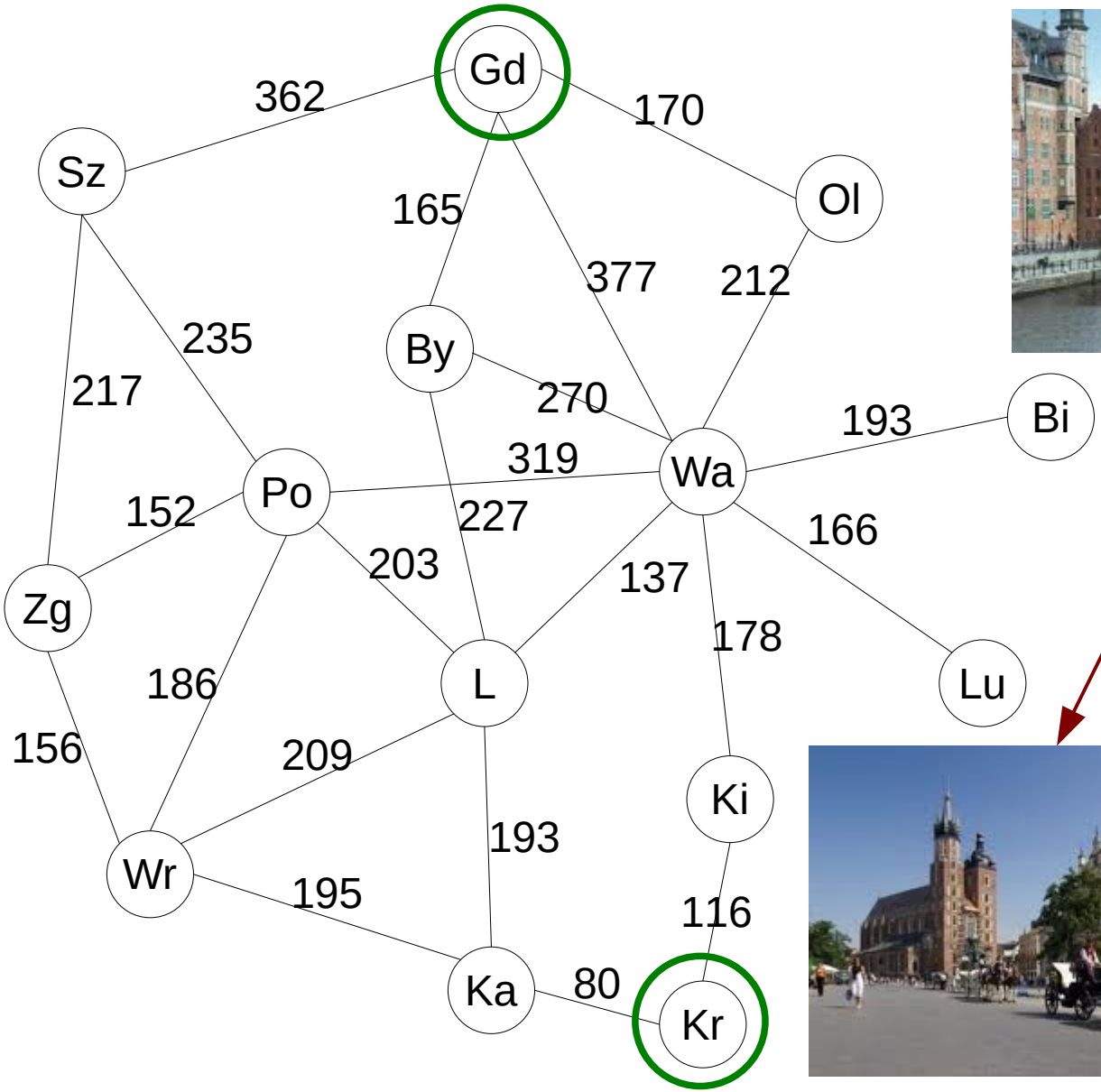
- Jeśli funkcja heurystyczna jest dopuszczalna i monotoniczna, to można zmniejszyć liczbę rozwijanych węzłów
- Algorytm musi wyczerpać listę A aby dać gwarancję uzyskania wyniku optymalnego

Algorytm A*

Algorytm najpierw najlepszy
+
Funkcja kosztu
+
Funkcja heurystyczna
=
Algorytm A*


Gd
Sz
By
Ol
Po
Wa
Bi
Zg
L
Wr
Ki
Lu
Ka
Kr
362
165
170
377
212
235
270
193
319
217
227
152
203
137
166
186
178
209
156
193
195
116
80

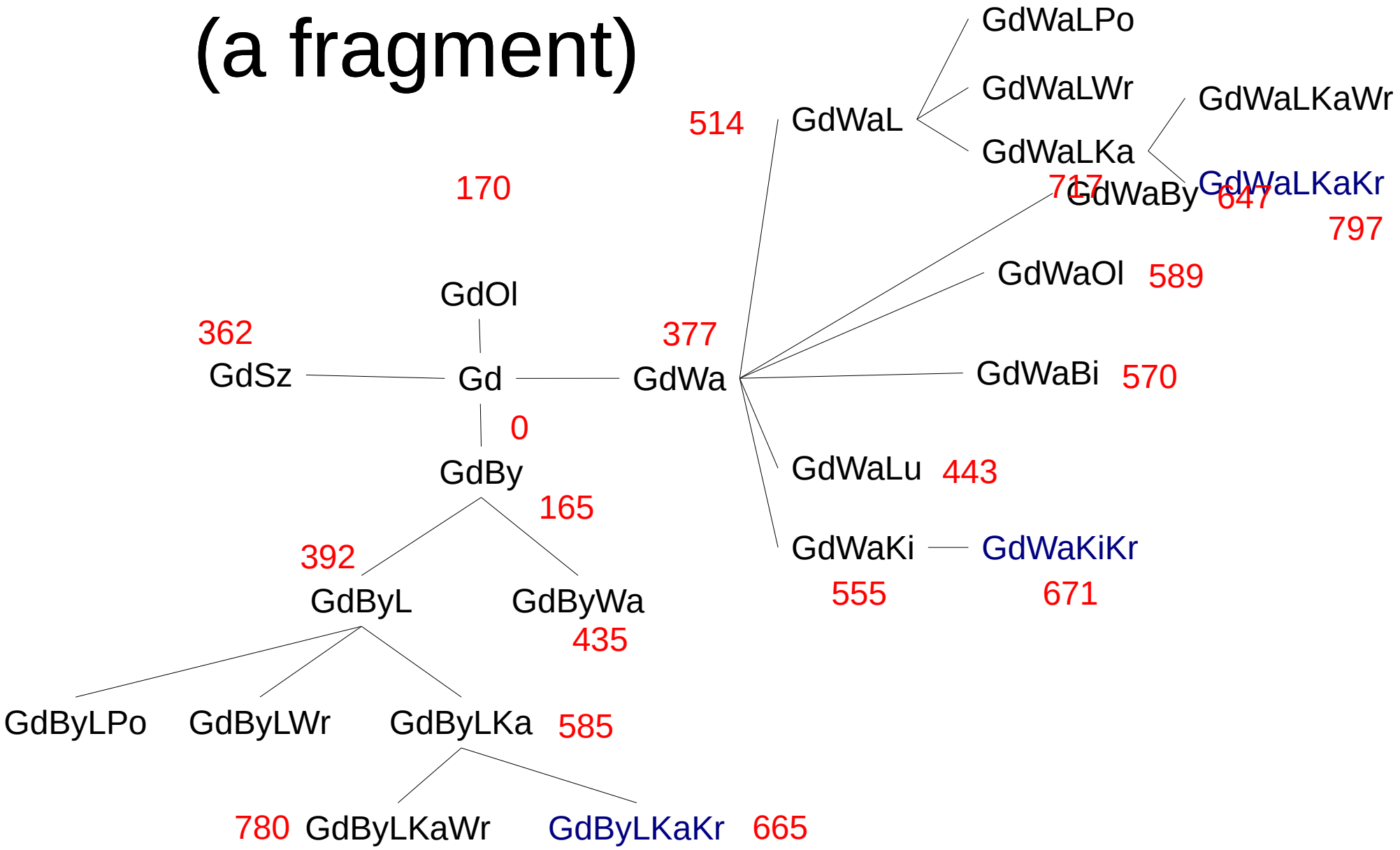
Find the shortest path from Gd to Kr



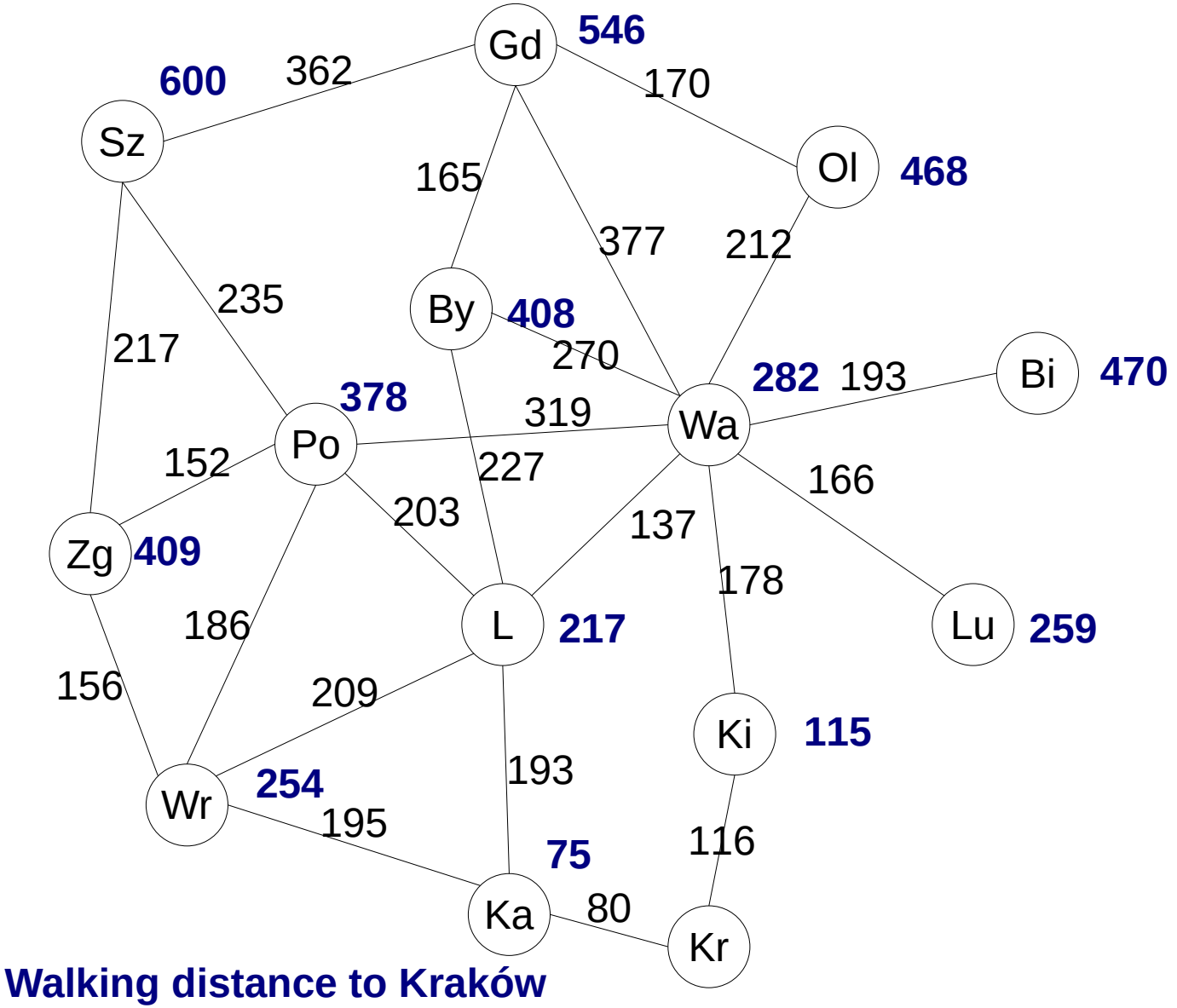
Gdańsk – old port

Kraków – market square

Search tree (a fragment)



Heuristic function – walking distane



Search tree (a fragment)