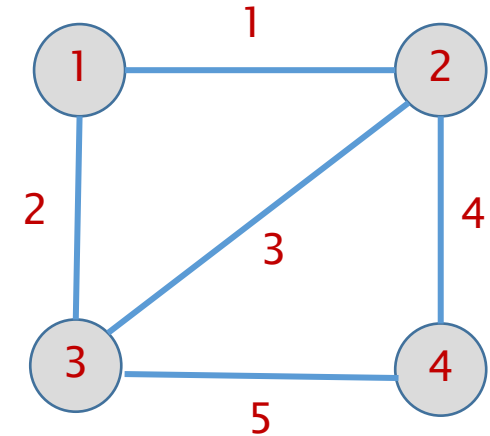
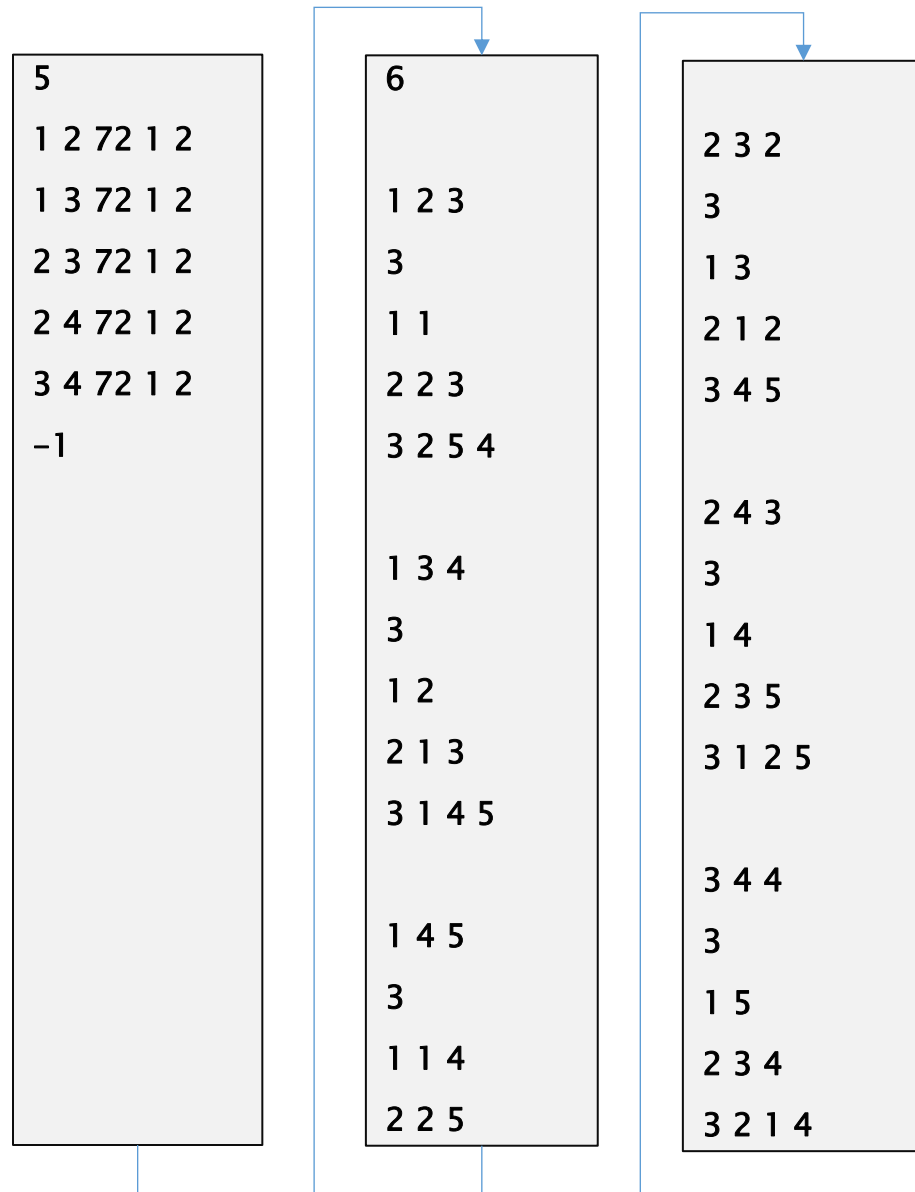


format danych

E
 $a(1) b(1) c(1) \xi(1) M(1)$
 $a(2) b(2) c(2) \xi(2) M(2)$
...
 $a(E) b(E) c(E) \xi(E) M(E)$
-1
 D
 $o(1) t(1) h(1)$
 $r(1)$
 $P(1,1) e1, e2,$
 $P(1,2) e1, e2, ...$
...
 $P(1, r(1)) e1, e2, ...$
 $o(2) t(2) h(2)$
 $r(2)$
 $P(2,1) e1, 2, ...$
 $P(2,2) e1, e2, ...$
...
 $P(2, r(2)) e1, e2, ...$

...
 $o(D) t(D) h(D)$
 $r(D)$
 $P(D,1) e1, e2, ...$
 $P(D,2) e1, e2, ...$
...
 $P(D, r(D)) e1, e2, ...$

format MP2k (net4.txt)



$h(d)$: 3, 4, 5, 2, 3, 4

d : (1,2), (1,3), (1,4), (2,3), (2,4), (3,4)

routes

$P(1,1) = \{1\}$, $P(1,2) = \{2,3\}$, $P(1,3) = \{2,4,5\}$

$P(2,1) = \{2\}$, $P(2,2) = \{1,3\}$, $P(2,3) = \{1,4,5\}$

$P(3,1) = \{1,4\}$, $P(3,2) = \{2,5\}$

$P(4,1) = \{3\}$, $P(4,2) = \{1,2\}$, $P(3,3) = \{4,5\}$

$P(5,1) = \{4\}$, $P(5,2) = \{3,5\}$, $P(5,3) = \{1,2,5\}$

$P(6,1) = \{5\}$, $P(6,2) = \{3,4\}$, $P(6,3) = \{1,2,4\}$

format MP2k - BNF (Backus-Naur Form)

`<network> ::= <links><EOL><separator><EOL><demands>`

`<separator> ::= "-1"`

`<links> ::= <number of links><EOL><link list>`

`<number of links> ::= <integer>`

`<linkList> := <link>[<EOL><link>]*`

`<link> ::= <start node> <end node> <number of modules> <module cost> <link module>`

`<start node> ::= <node id>`

`<end node> ::= <node id>`

`<number of modules> ::= <integer>`

`<module cost> ::= <float>`

`<link module> ::= <integer>`

`<node id> ::= <integer>`

5

1 2 72 1 2

1 3 72 1 2

2 3 72 1 2

2 4 72 1 2

3 4 72 1 2

-1

Notacja ta jest powszechnie używana w informatyce do zapisu składni (syntaktyki) języków programowania i protokołów komunikacyjnych. Została wymyślona przez Johna Backusa w latach 50. w czasie prac nad językiem Fortran, a następnie zmodyfikowana przez Petera Naura i użyta do zdefiniowania składni języka Algol (z Wikipedii).

format MP2k - BNF (Backus-Naur Form) - 2

`<demands> ::= <number of demands><EOL><demand list>`

`<number of demands> ::= <integer>`

`<demand list> ::= <demand>[<EOL><demand>]*`

`<demand> ::= <start node> <end node> <demand volume><EOL><demand paths>`

`<demand volume> ::= <integer>`

`<demand paths> ::= <number of demand paths><EOL><demand path list>`

`<number of demand paths> ::= <integer>`

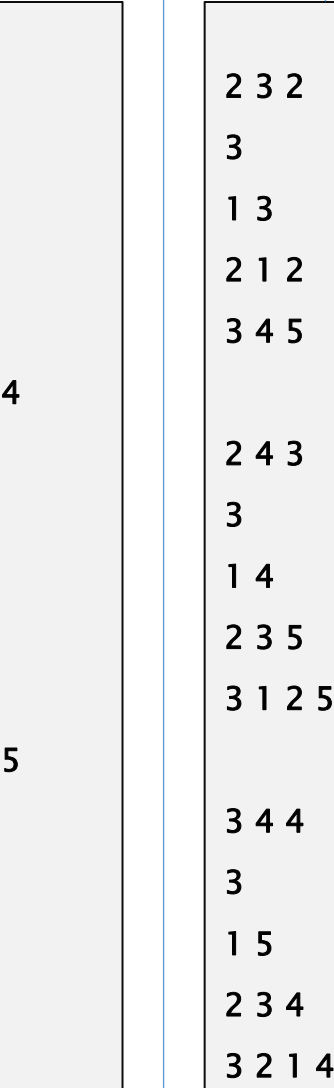
`<demand path list> ::= <demand path>[<EOL><demand path>]*`

`<demand path> ::= <demand path id> <link list><EOL>`

`<demand path id> ::= <integer>`

`<link list> ::= <link id>[<link id>]*`

`<link id> ::= <integer>`



6	2 3 2
1 2 3	3
3	1 3
1 1	2 1 2
2 2 3	3 4 5
3 2 5 4	
	2 4 3
1 3 4	3
3	1 4
1 2	2 3 5
2 1 3	3 1 2 5
3 1 4 5	
	3 4 4
1 4 5	3
3	1 5
1 1 4	2 3 4
2 2 5	3 2 1 4

polska network (net12_1.txt, net12_2.txt)

