

Programovacie techniky

Binárne vyhľadávanie

a indexovanie

Vladislav Novák

Hľadanie prekladu v slovníku



potato: zemiak

rice: ryža

Ako urýchliť
preklad?

apple: jablko

broccoli: brokolica

pepper: paprika

cucumber: uhorka

carrot: mrkva

orange: pomaranč

cabbage: kapusta

banana: banán

leek: pór

Hľadanie prekladu v slovníku



A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

Z

rice: ryža

Napríklad
usporadáním

Hľadanie prekladu v slovníku



rice

A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

Z

rice: ryža

Kde začneme
hľadať?

Hľadanie prekladu v slovníku



rice

A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

Z

rice: ryža

Kde začneme
hľadať?

Od stredu

Hľadanie prekladu v slovníku



rice

Po 1. porovnaní
stačí prehľadať len
polovicu slovníka

A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

Z

rice: ryža

Kde začneme
hľadať?

Od stredu

Hľadanie prekladu v slovníku



rice

Po 1. porovnaní
stačí prehľadať len
polovicu slovníka

Po 2. porovnaní
len štvrtinu, atď.

A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

rice: ryža

Z

Kde začneme
hľadať?

Od stredu

Hľadanie prekladu v slovníku



rice

Po 1. porovnaní
stačí prehľadať len
polovicu slovníka

Po 2. porovnaní
len štvrtinu, atď.

A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

rice: ryža

Z

Kde začneme
hľadať?

Od stredu

Binárne
vyhľadávanie

Hľadanie prekladu v slovníku



rice

A

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

rice: ryža

Z

Kde začneme
hľadať?

Od stredu

Po 1. porovnaní
stačí prehľadať len
polovicu slovníka

Po 2. porovnaní
len štvrtinu, atď.

Slovník nemusí
obsahovať preklad

Binárne
vyhľadávanie

Hľadanie prekladu v slovníku



rice

Po 1. porovnaní
stačí prehľadať len
polovicu slovníka

Po 2. porovnaní
len štvrtinu, atď.

Slovník nemusí
obsahovať preklad

1	A
2	B
3	C
4	D
5	E
6	F
7	G
8	H
9	I
10	J
11	K
12	L
13	M
14	N
15	O
16	P
17	Q
18	R
19	S
20	T
21	U
22	V
23	W
24	X
25	Y
26	Z

A

Z

apple: jablko

banana: banán

broccoli: brokolica

cabbage: kapusta

carrot: mrkva

cucumber: uhorka

leek: pór

orange: pomaranč

potato: zemiak

pepper: paprika

rice: ryža

Kde začneme
hľadať?

Od stredu

Ak nie sú k
dispozícii
ďalšie
užitočné
informácie

Binárne
vyhľadávanie

Binárne hľadanie



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Binárne hľadanie

Hodnoty sú usporiadané



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Binárne hľadanie

Hodnoty sú usporiadané

Rozsah hodnôt 5 až 200,
ale hodnota 100 je takmer úplne vpravo

a hodnota 50 je vpravo od stredu



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Binárne hľadanie

Hodnoty sú usporiadané

Rozsah hodnôt 5 až 200,
ale hodnota 100 je takmer úplne vpravo

a hodnota 50 je vpravo od stredu



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Bez dodatočných informácií
nevieme odhadnúť
kde sa zadaná hodnota
môže nachádzať

Binárne hľadanie



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Binárne hľadanie



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Binárne hľadanie



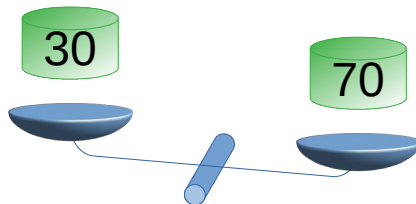
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17



Binárne hľadanie




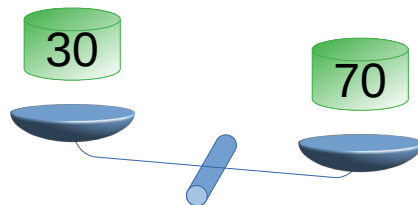
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17



Binárne hľadanie



																	
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17



Podľa hodnoty v poli viem,
či treba hľadať vľavo alebo vpravo

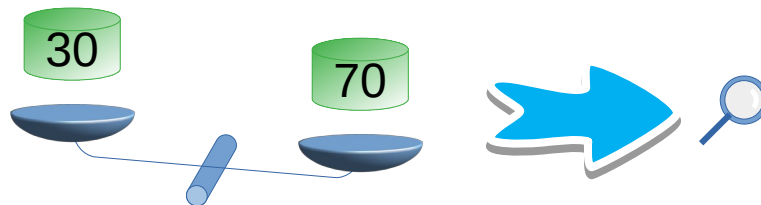


Binárne hľadanie



5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

17




Binárne hľadanie

Implementácia:

- iteratívna
- rekurzívna

Binárne hľadanie

Implementácia:

- iteratívna 
- rekurzívna

Binárne hľadanie

5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Binárne hľadanie

5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

```
binarySearch(array, value) -> index|notFound {
```

```
    return notFound  
}
```

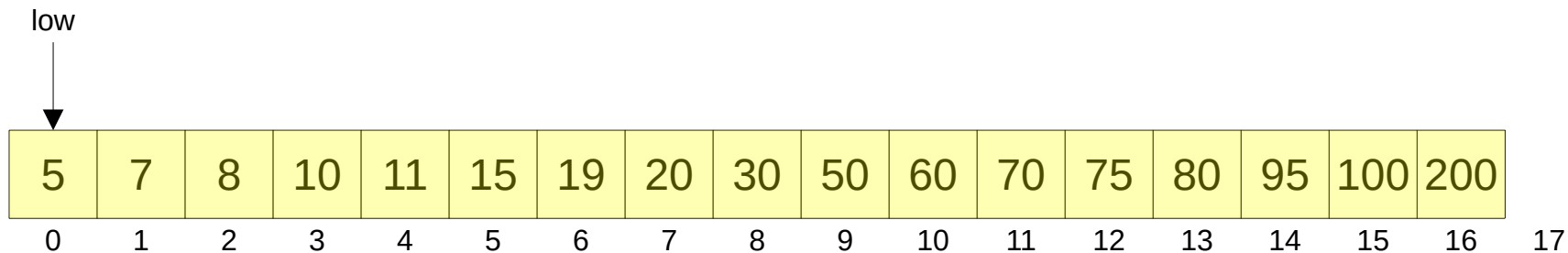

Binárne hľadanie

5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

```
binarySearch(array, value) -> index|notFound { value = 70
```

```
    return notFound  
}
```

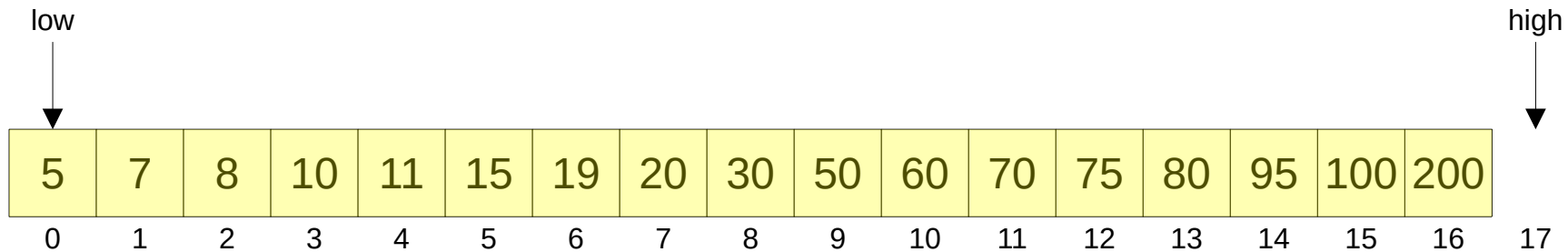
Binárne hľadanie



```
binarySearch(array, value) -> index|notFound { value = 70  
  → low = ..... // index 1. prvku  
    high = ..... // index na/za posledný prvok
```

```
    return notFound  
}
```

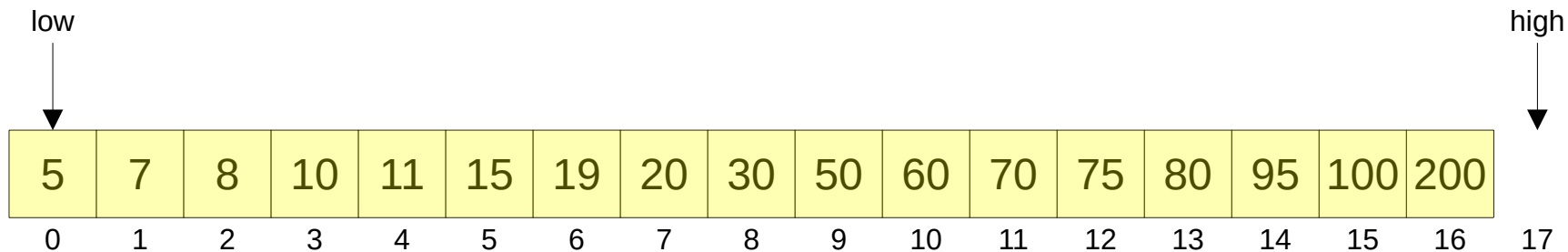
Binárne hľadanie



```
binarySearch(array, value) -> index|notFound { value = 70  
  low = ..... // index 1. prvku  
  ➔ high = ..... // index na/za posledný prvok
```

```
    return notFound  
}
```

Binárne hľadanie

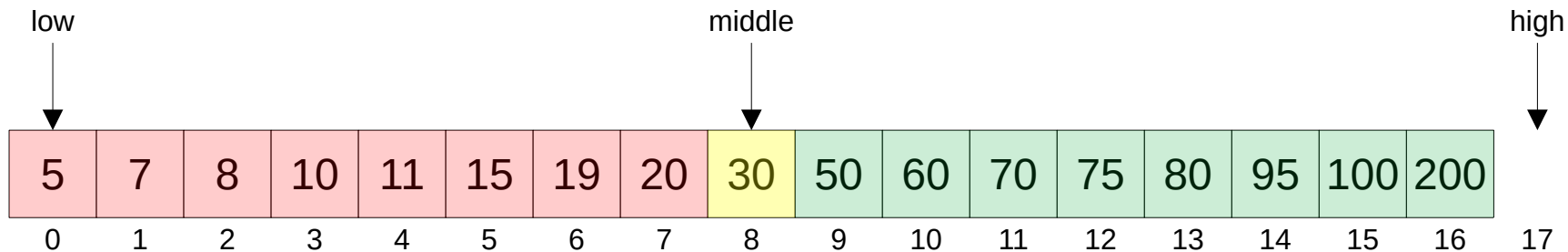


```
binarySearch(array, value) -> index|notFound {  
    low = ..... // index 1. prvku  
    high = ..... // index na/za posledný prvok
```

```
    ➔ loop(ak je v intervale low až high aspoň jeden prvok) {
```

```
    }  
    return notFound  
}
```

Binárne hľadanie

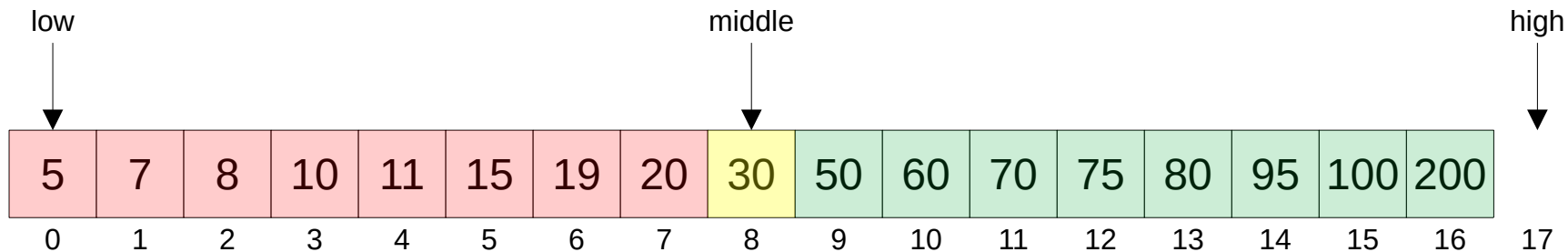


```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    ➔ middle = ..... // vypočítaj stred

  }
  return notFound
}
```

Binárne hľadanie

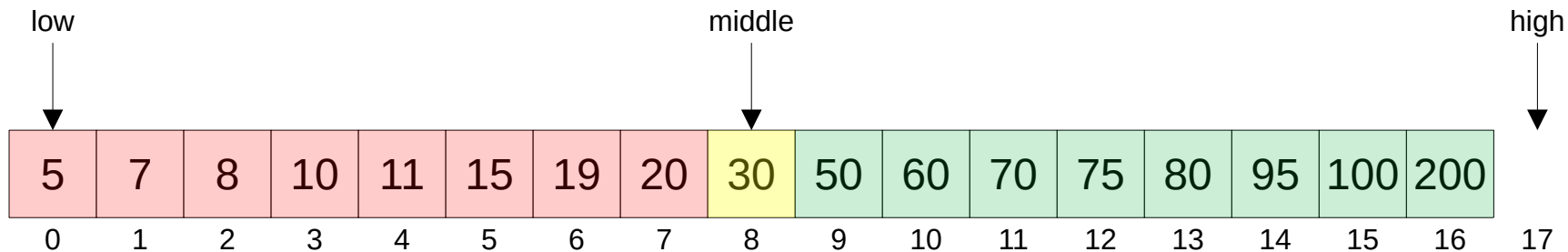


```
binarySearch(array, value) -> index|notFound {  
    low = ..... // index 1. prvku  
    high = ..... // index na/za posledný prvok  
  
    loop(ak je v intervale low až high aspoň jeden prvok) {  
        middle = ..... // vypočítaj stred  
        → if(value == array[middle]) {  
            return middle  
        }  
    }  
    return notFound  
}
```

Porovnávané prvky:

30

Binárne hľadanie

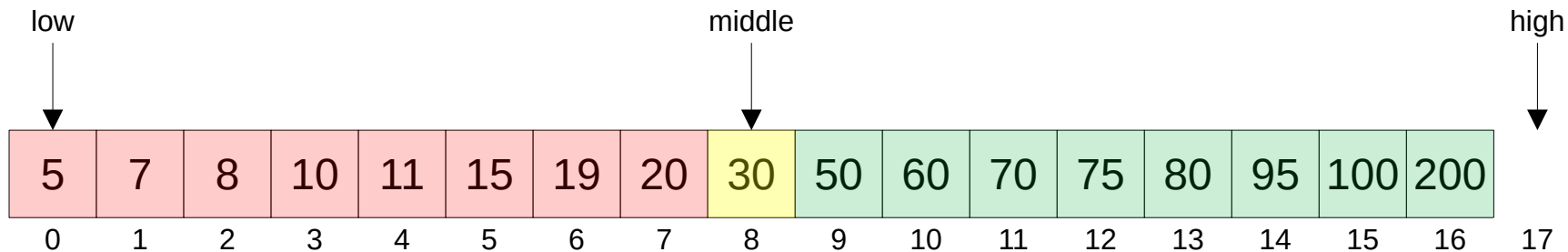


```
binarySearch(array, value) -> index|notFound {  
    low = ..... // index 1. prvku  
    high = ..... // index na/za posledný prvok  
  
    loop(ak je v intervale low až high aspoň jeden prvok) {  
        middle = ..... // vypočítaj stred  
        if(value == array[middle]) {  
            return middle  
        }  
        → if(value < array[middle]) {  
            high = ..... // posuň hornú hranicu  
        }  
        if(value > array[middle]) {  
            low = ..... // posuň dolnú hranicu  
        }  
    }  
    return notFound  
}
```

Porovnávané prvky:

30

Binárne hľadanie



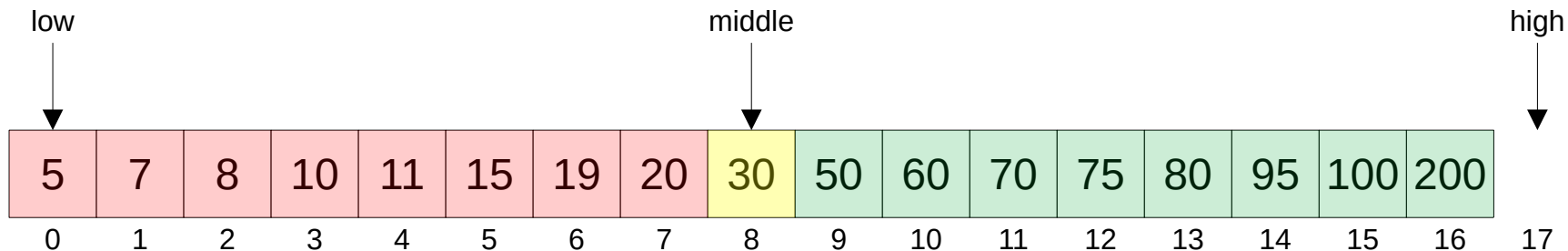
```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    middle = ..... // vypočítaj stred
    if(value == array[middle]) {
      return middle
    }
    if(value < array[middle]) {
      high = ..... // posuň hornú hranicu
    }
    → if(value > array[middle]) { if (70 > 30)
      low = ..... // posuň dolnú hranicu
    }
  }
  return notFound
}
```

Porovnávané prvky:

30

Binárne hľadanie



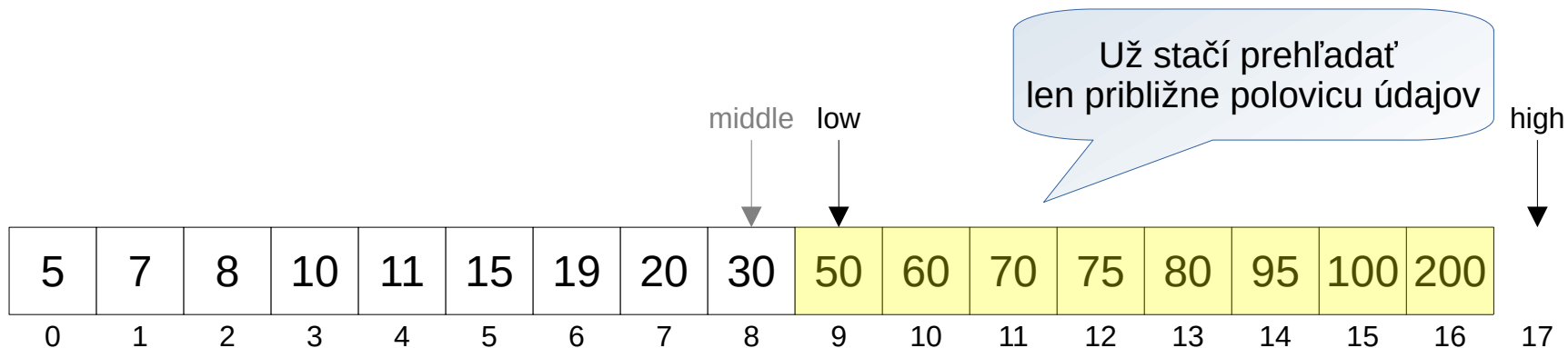
```
binarySearch(array, value) -> index|notFound {  
    low = ..... // index 1. prvku  
    high = ..... // index na/za posledný prvok  
  
    loop(ak je v intervale low až high aspoň jeden prvok) {  
        middle = ..... // vypočítaj stred  
        if(value == array[middle]) {  
            return middle  
        }  
        if(value < array[middle]) {  
            high = ..... // posuň hornú hranicu  
        }  
        → if(value > array[middle]) {  
            low = ..... // posuň dolnú hranicu  
        }  
    }  
    return notFound  
}
```

else

Porovnávané prvky:

30

Binárne hľadanie



```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    loop(ak je v intervale low až high aspoň jeden prvok) {
        middle = ..... // vypočítaj stred
        if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            ➔ low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30

Binárne hľadanie

								middle	low									high
								↓	↓									↓
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

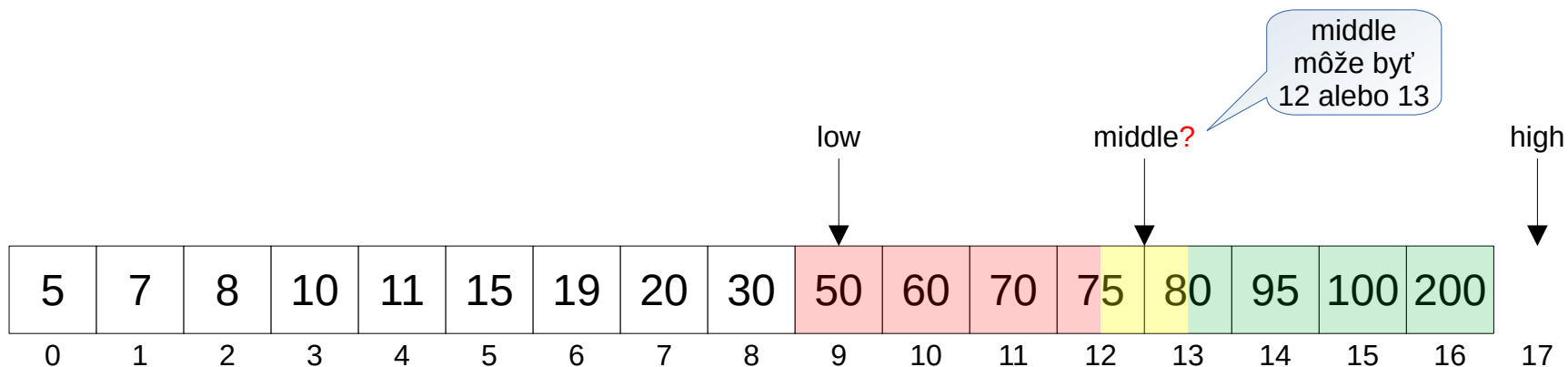
```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    → loop(ak je v intervale low až high aspoň jeden prvok) {
        middle = ..... // vypočítaj stred
        if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30

Binárne hľadanie



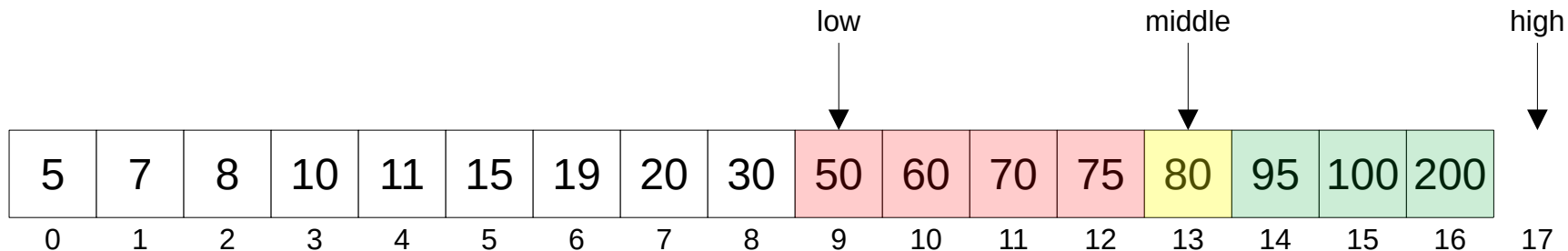
```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    loop(ak je v intervale low až high aspoň jeden prvok) {
        ➔ middle = ..... // vypočítaj stred
        if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30

Binárne hľadanie



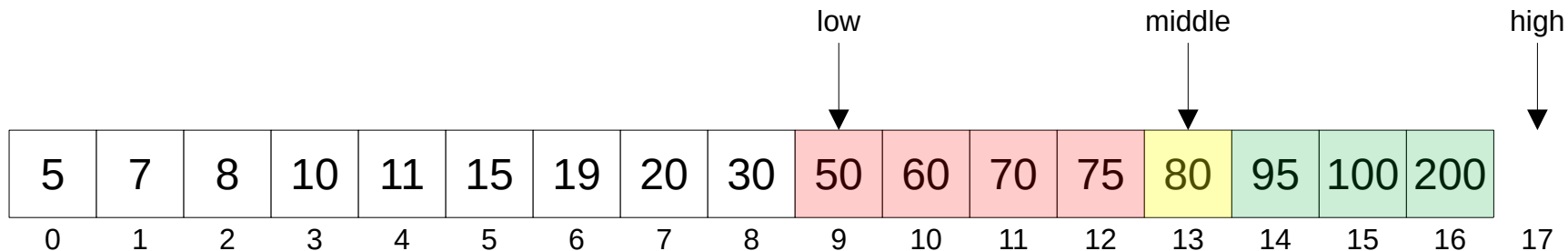
```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    loop(ak je v intervale low až high aspoň jeden prvok) {
        ➔ middle = ..... // vypočítaj stred
        if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30

Binárne hľadanie



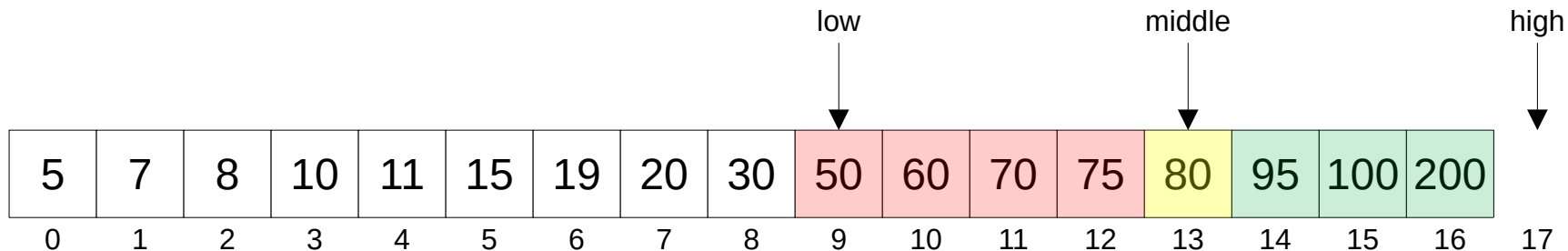
```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    loop(ak je v intervale low až high aspoň jeden prvok) {
        middle = ..... // vypočítaj stred
        ➔ if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30
80

Binárne hľadanie



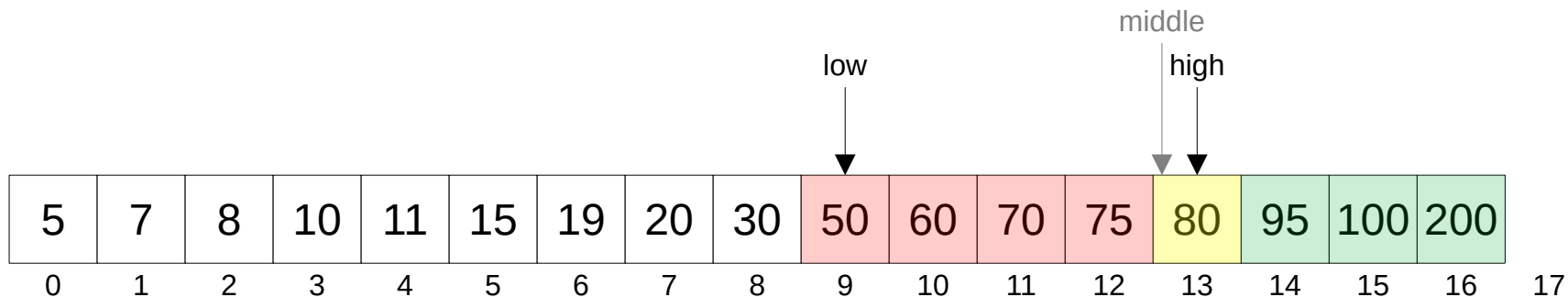
```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    middle = ..... // vypočítaj stred
    if(value == array[middle]) {
      return middle
    }
    → if(value < array[middle]) {
      high = ..... // posuň hornú hranicu
    }
    if(value > array[middle]) {
      low = ..... // posuň dolnú hranicu
    }
  }
  return notFound
}
```

Porovnávané prvky:

30
80

Binárne hľadanie



```
binarySearch(array, value) -> index|notFound {  
    low = ..... // index 1. prvku  
    high = ..... // index na/za posledný prvok  
  
    loop(ak je v intervale low až high aspoň jeden prvok) {  
        middle = ..... // vypočítaj stred  
        if(value == array[middle]) {  
            return middle  
        }  
        if(value < array[middle]) {  
            ➔ high = ..... // posuň hornú hranicu  
        }  
        if(value > array[middle]) {  
            low = ..... // posuň dolnú hranicu  
        }  
    }  
    return notFound  
}
```

Porovnávané prvky:

30
80

Binárne hľadanie

									low		middle		high					
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

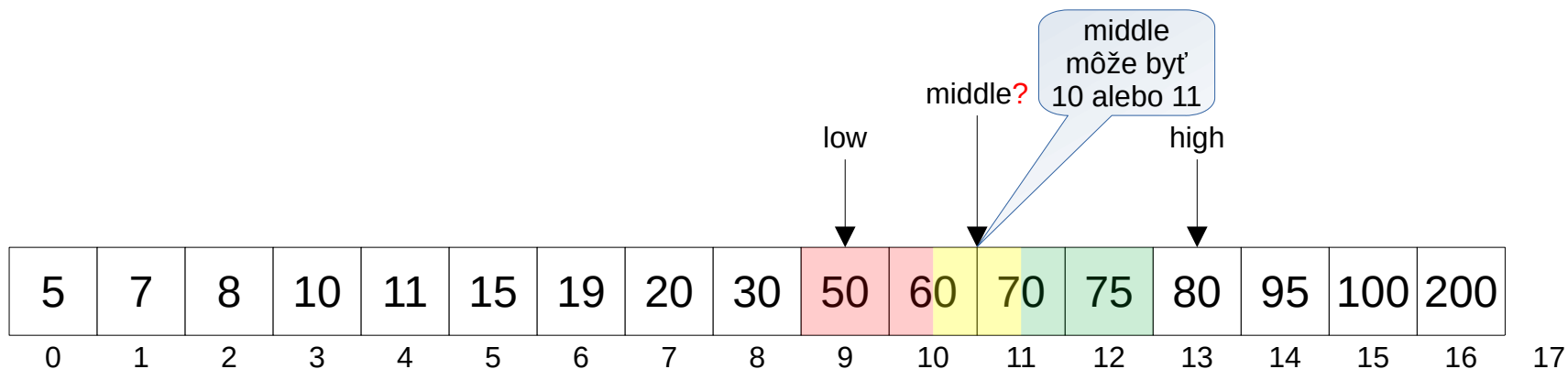
```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    → loop(ak je v intervale low až high aspoň jeden prvok) {
        middle = ..... // vypočítaj stred
        if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30
80

Binárne hľadanie



```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    ➔ middle = ..... // vypočítaj stred
    if(value == array[middle]) {
      return middle
    }
    if(value < array[middle]) {
      high = ..... // posuň hornú hranicu
    }
    if(value > array[middle]) {
      low = ..... // posuň dolnú hranicu
    }
  }
  return notFound
}
```

Porovnávané prvky:

30
80

Binárne hľadanie

									low	middle		high						
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

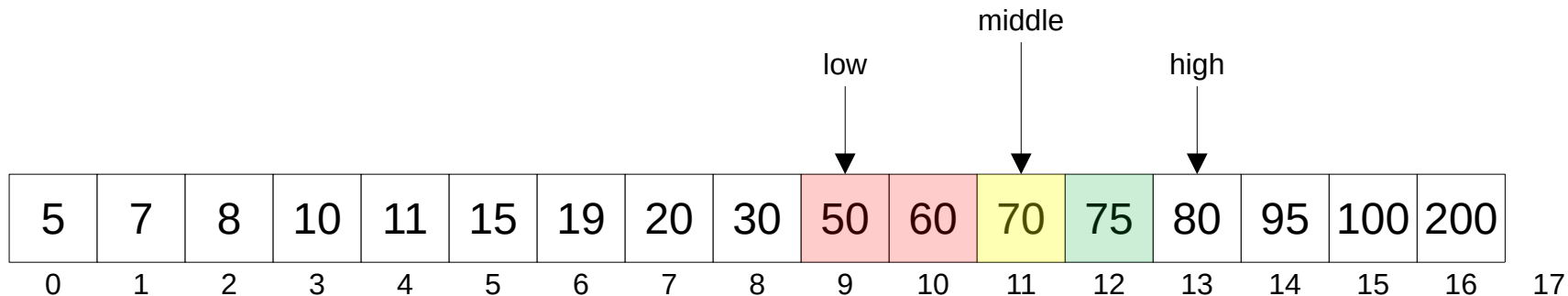
```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    ➔ middle = ..... // vypočítaj stred
    if(value == array[middle]) {
      return middle
    }
    if(value < array[middle]) {
      high = ..... // posuň hornú hranicu
    }
    if(value > array[middle]) {
      low = ..... // posuň dolnú hranicu
    }
  }
  return notFound
}
```

Porovnávané prvky:

30
80

Binárne hľadanie



```
binarySearch(array, value) -> index|notFound { value = 70
    low = ..... // index 1. prvku
    high = ..... // index na/za posledný prvok

    loop(ak je v intervale low až high aspoň jeden prvok) {
        middle = ..... // vypočítaj stred
        ➔ if(value == array[middle]) {
            return middle
        }
        if(value < array[middle]) {
            high = ..... // posuň hornú hranicu
        }
        if(value > array[middle]) {
            low = ..... // posuň dolnú hranicu
        }
    }
    return notFound
}
```

Porovnávané prvky:

30
80
70

Binárne hľadanie

										low		middle		high			
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200	
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    middle = ..... // vypočítaj stred
    if(value == array[middle]) {
      ➔ return middle
    }
    if(value < array[middle]) {
      high = ..... // posuň hornú hranicu
    }
    if(value > array[middle]) {
      low = ..... // posuň dolnú hranicu
    }
  }
  return notFound
}
```

Porovnávané prvky:

30
80
70

Binárne hľadanie

										low		middle		high				
5	7	8	10	11	15	19	20	30	50	60	70	75	80	95	100	200		
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

```
binarySearch(array, value) -> index|notFound { value = 70
  low = ..... // index 1. prvku
  high = ..... // index na/za posledný prvok

  loop(ak je v intervale low až high aspoň jeden prvok) {
    middle = ..... // vypočítaj stred
    if(value == array[middle]) {
      ➔ return middle
    }
    if(value < array[middle]) {
      high = ..... // posuň hornú hranicu
    }
    if(value > array[middle]) {
      low = ..... // posuň dolnú hranicu
    }
  }
  return notFound
}
```

Porovnávané prvky:

30
80
70

Príklad použitia

- binárne hľadanie
- indexovanie

Databáza študentov



Databáza študentov



id : 1001
meno: Tibor
o.p.: JI05
roč.: 1



id : 1007
meno: Andrej
o.p.: BT01
roč.: 3



id : 1008
meno: Peter
o.p.: EV03
roč.: 1



id : 1009
meno: Andrej
o.p.: DW02
roč.: 1



id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Popis skratiek:

id – identifikačné číslo
meno – meno
o.p. – číslo občianskeho preukazu
roč. – aktuálny ročník štúdia

Podotázka:

aký údaj by bolo lepšie zaznamenávať
namiesto aktuálneho ročníka štúdia?

Databáza študentov

Hľadanie informácií o študentovi podľa ID



id : 1001
meno: Tibor
o.p.: JI05
roč.: 1



id : 1007
meno: Andrej
o.p.: BT01
roč.: 3



id : 1008
meno: Peter
o.p.: EV03
roč.: 1



id : 1009
meno: Andrej
o.p.: DW02
roč.: 1



id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Popis skratiek:

id – identifikačné číslo
meno – meno
o.p. – číslo občianskeho preukazu
roč. – aktuálny ročník štúdia

Podotázka:

aký údaj by bolo lepšie zaznamenávať
namiesto aktuálneho ročníka štúdia?

Databáza študentov

Hľadanie informácií o študentovi podľa ID

Predpokladáme, že nevieme určiť pozíciu údajov o študentovi priamo z ID.

Napríklad ID nie je indexom v poli informácií o študentoch.

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Databáza študentov

Hľadanie informácií o študentovi podľa ID

Predpokladáme, že nevieme určiť pozíciu údajov o študentovi priamo z ID.

Napríklad ID nie je indexom v poli informácií o študentoch.

usporiadanie
podľa id

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Databáza študentov

Hľadanie informácií o študentovi podľa ID

Môžeme využiť binárne hľadanie

Predpokladáme, že nevieme určiť pozíciu údajov o študentovi priamo z ID.

Napríklad ID nie je indexom v poli informácií o študentoch.

usporiadanie
podľa id

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Databáza študentov

Hľadanie informácií o študentovi podľa občianskeho preukazu

usporiadanie
podľa id

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Databáza študentov

Hľadanie informácií o študentovi podľa občianskeho preukazu

Záznamy nie sú usporiadané
podľa občianskych preukazov

usporiadanie
podľa id

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Databáza študentov

Hľadanie informácií o študentovi podľa občianskeho preukazu

Vytvoríme index

Záznamy nie sú usporiadané
podľa občianskych preukazov

usporiadanie
podľa id

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

Databáza študentov

Hľadanie informácií o študentovi podľa občianskeho preukazu

Vytvoríme index

Záznamy nie sú usporiadané
podľa občianskych preukazov

usporiadanie
podľa id

id : 1001
meno: Tibor
o.p.: JI05
roč.: 1

id : 1007
meno: Andrej
o.p.: BT01
roč.: 3

id : 1008
meno: Peter
o.p.: EV03
roč.: 1

id : 1009
meno: Andrej
o.p.: DW02
roč.: 1

id : 1010
meno: Miro
o.p.: PT04
roč.: 2

usporiadanie
podľa občianskeho preukazu

o.p.: BT01

o.p.: DW02

o.p.: EV03

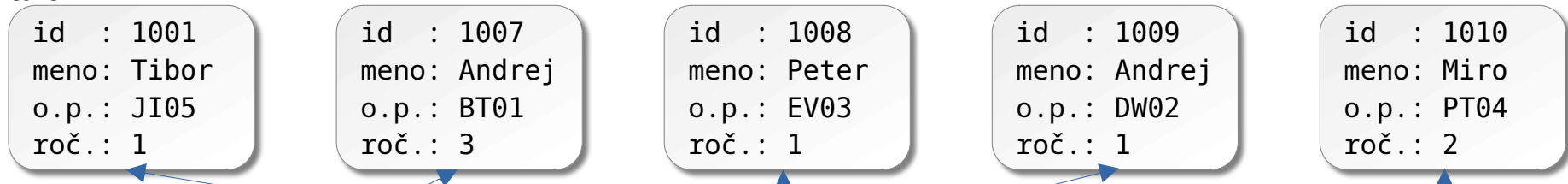
o.p.: JI05

o.p.: PT04

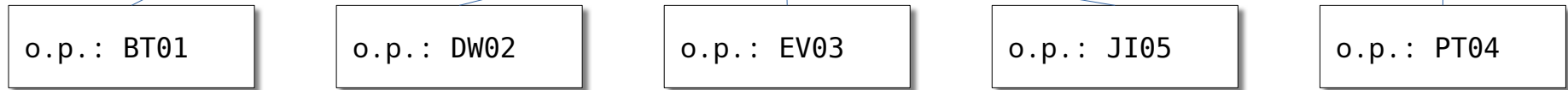
Databáza študentov

Hľadanie informácií o študentovi podľa mena

usporiadanie
podľa id



usporiadanie
podľa občianskeho preukazu



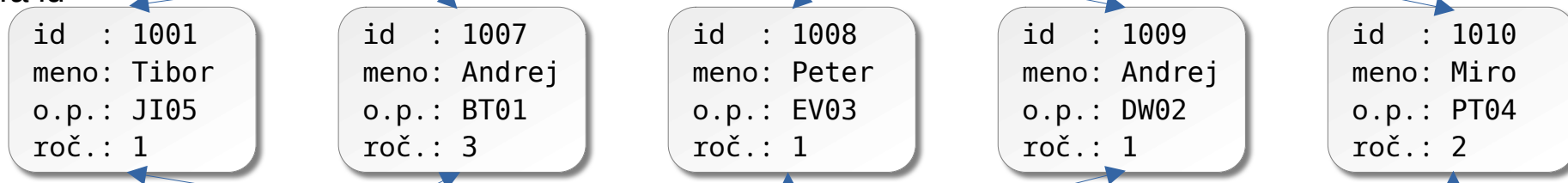
Databáza študentov

Hľadanie informácií o študentovi podľa mena

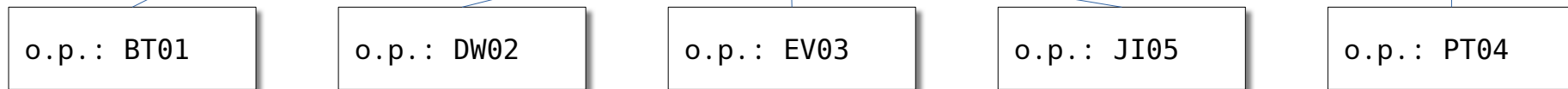
usporiadanie
podľa mena



usporiadanie
podľa id



usporiadanie
podľa občianskeho preukazu



Databáza študentov

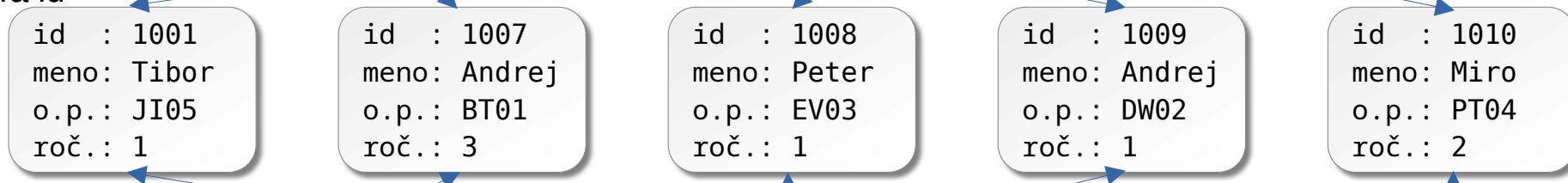
Hľadanie informácií o študentovi podľa mena

Viacerí študenti môžu mať rovnaké meno

usporiadanie
podľa mena



usporiadanie
podľa id



usporiadanie
podľa občianskeho preukazu

