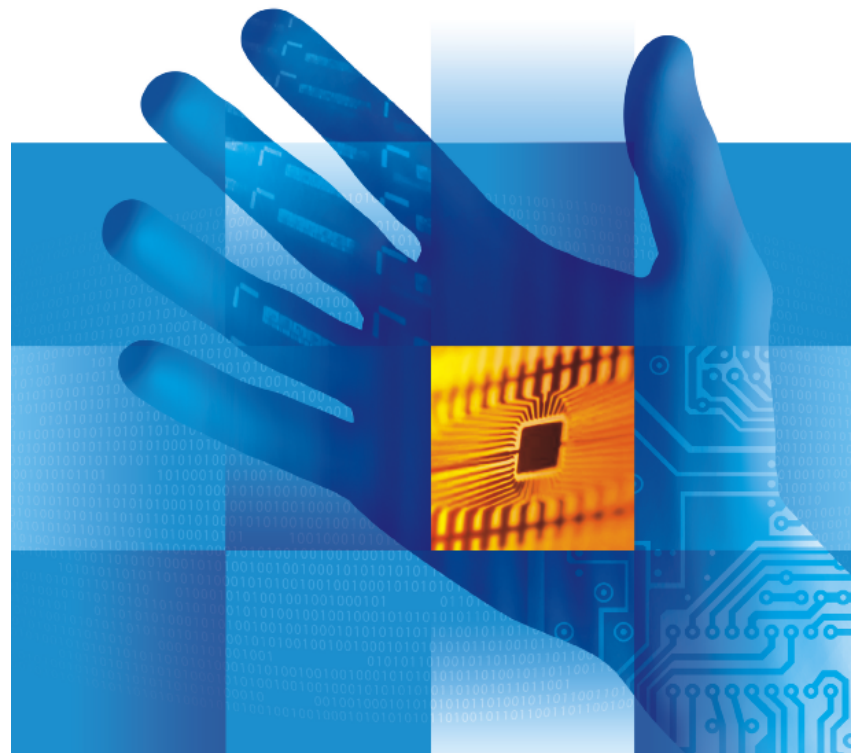




Cvičení/Lab 10

poznámky/notes

doc. RNDr. Petr Šaloun, Ph.D.
VŠB-TUO, FEI



průnik množin

alokace množiny **A** s **N** prvky

```
int *arrayA = new int[N];
```

nutnost průběžně kontrolovat nový prvek

```
bool isInSet(int item, int array[], int size){
```

průnik lze průběžně tisknout při čtení B

pozor na výstup

```
{ } versus { , }
```

(kdy tisknout: , hodnota -- logická

proměnná pomůže)



Sets Intersection

allocation of set **A** with **N** elements

```
int *arrayA = new int[N];
```

check presence of a new element

```
bool isInSet(int item, int arry[], int size){
```

print the intersection concurrently as
you read elements from set **B**

teke care for mistake

```
{ } versus { , }
```

(when print: , element -- employ an
boolean variable)



Kosinová míra podobnosti

N počet prvků vektoru

```
double *vctr1 = new double[N];
```

```
// číst a kontrolovat nesprávný vstup
```

```
// totéž pro vctr2
```

kontroluje se uvolnění paměti

```
// delete [] vctr1;
```

```
// a to i po nesprávném vstupu
```

pak výpočet

```
// ideálně funkce pro  $|x|$ ,  $|y|$  a  $|x * y|$ 
```

```
// velikost(x)...
```

```
// ale jde i v jediném cyklu
```





Cosine Similarity

N number of elements in vector
double *vctr1 = new double[N];

// allocate memory

// read vctr1 and check if fail

// repeat the same for vctr2

return allocated memory back

// delete [] vctr1;

// even, if input fails

then evaluate

// functions for $|x|$, $|y|$ a $|x * y|$ are welcome

// scalar_product(x)... vector_product()

// but one loop solve it as well

