

# EADŠ - cvičenie 5

21. októbra 2022

## Načítavanie vstupu zo súboru

Ak ste niekedy robili s príkazovým riadkom pomerne priamočiare:

```
python du1-3.py < ./test/1.a.in
```

Vyhnete sa problémom s načítavaním vstupu.

## Vypisovanie výstupu do súboru

Ak ste niekedy robili s príkazovým riadkom tiež pomerne priamočiare:

```
python du1-3.py < ./test/1.a.in > ./test/1.a.out
```

Takto máte súbor s výstupom a viete napríklad porovnávať dve riešenia.

# Odpovede testovača

our 0001: 7\_10\_5\_10

your 0001: 4\_10\_2\_10

# Odpovede testovača

```
your terminated, our continues:  
our 0001: 0_0_0_0
```

# Odpovede testovača

```
our terminated, your continues:  
your 0001:
```

## Opakovanie - vyhľadavanie v texte

Naivne: pre každý index vyskúšame či tam môže byť slovo (zložitosť  $O(nm)$ ).

```
for i in range(len(N)):
    j=0
    while j<len(M):
        if i+j>=len(N):
            break
        if N[i+j] != M[j]:
            break
        j += 1

    if j==len(M):
        print(f"nasiel som, zaciatok je na indexe {i}")
```

Vymyslite príklad, kedy je zložitosť naozaj  $\Theta(nm)$ ?

# Opakovanie - vyhľadavanie v texte - hashovanie

Chceli by sme niečo lepšie.

Aktuálne: pre každé  $n$  robíme  $m$  porovnaní (dva stringy dĺžky  $m$ ).  
Bolo by fajn, ak by sme vedeli porovnávať stringy v  $O(1)$ .

To ale možno vieme - hashovanie.

Problém: hashovanie zaberá  $O(m)$ .

Ako prehashovať v  $O(1)$ ? Rolling hash.



## Opakovanie - rolling hash

Nech  $p$  je prvočíslo a  $w$  je reťazec

$$H(w[i : i+k]) = w[i] \cdot p^{k-1} + w[i+1] \cdot p^{k-2} + \dots + w[i+k-1] \cdot p^0$$

potom

$$H(w[i+1 : i+k+1]) = w[i+1] \cdot p^{k-1} + \dots + w[i+k-1] \cdot p^1 + w[i+k] \cdot p^0$$

teda

$$H(w[i+1 : i+k+1]) = p \cdot (H(w[i : i+k]) - w[i] \cdot p^{k-1}) + w[i+k] \cdot p^0$$

# Opakovanie - vyhľadavanie v texte - KMP

Chceli by sme niečo lepšie.

Aktuálne: po nájdení rozdielnych znakov sa vraciame naspäť, a (možno) zbytočne. Bolo by fajn, ak by sme nemuseli vracat' vôbec.

Chceli by sme si predpočítať, aký je ďalší najlepší index v hľadanom slove.

# Vyhľadávanie v texte - 1

Máme dva stringy:  $u$  a  $v$ . Chceme zistiť, či je jeden rotáciou druhého.

Rotácie stringu  $abcd$ :

- ▶  $abcd$
- ▶  $bcda$
- ▶  $cdab$
- ▶  $dabc$
- ▶  $abcd$
- ▶ ...

# Suffixový strom - 1

Dostaneme string  $S$ . Chceme zistiť, aký je v ňom najdlhší opakujúci sa podreťazec.

## Suffixový strom - 2

Dostaneme dva stringy  $S$  a  $T$ . Chceme zistiť, aký je v nich najdlhší spoločný podreťazec.