Vyhledávání v textu

doc. Mgr. Jiří Dvorský, Ph.D.

Katedra informatiky
Fakulta elektrotechniky a informatiky
VŠB – TU Ostrava

Prezentace ke dni 12. února 2020







Osnova přednášky

Vyhledávání v textu

- Definice základních pojmů
- Vyhledávání hrubou silou
- Morris-Prattův algoritmus
- Knuth-Morris-Prattův algoritmus

Definice základních pojmů

Definice

Abeceda Σ je konečná neprázdná množina symbolů.

Příklady:

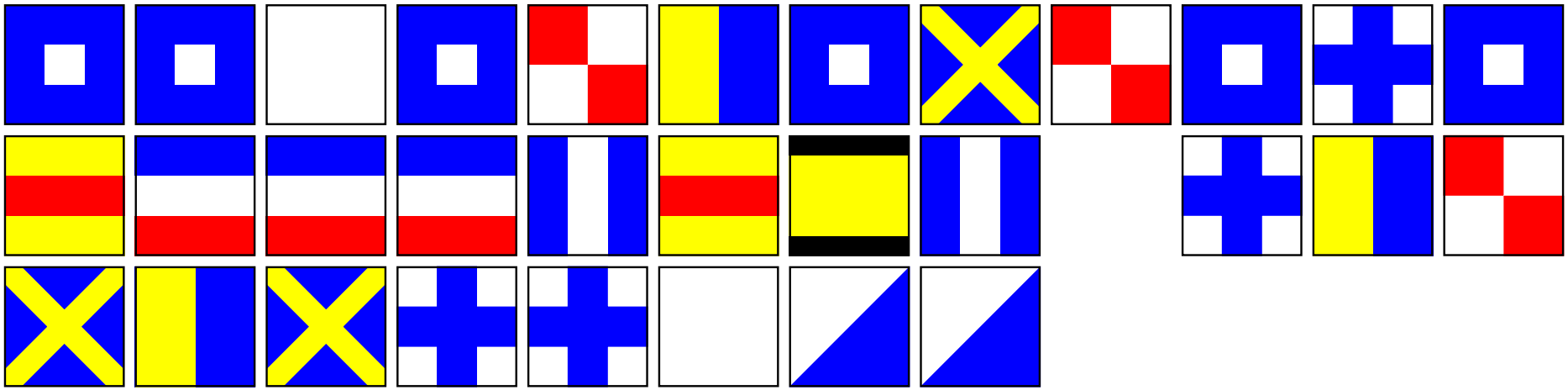
- binární abeceda $\Sigma = \{0, 1\}$,
- Morseova abeceda $\Sigma = \{., -, /\}$,
- abeceda aminokyselin DNA $\Sigma = \{A, C, G, T\}$,
- anglická abeceda $\Sigma = \{a, \ldots, z\}$.

Vlajková abeceda

- abeceda může mít i fyzickou podobu, viz například námořní vlajková abeceda, `en.wikipedia.org/wiki/International_maritime_signal_flags`
- současná podoba z roku 1931,
- celkem 40 vlajek – 26 písmen, 10 číslic, 1 potvrzovací vlajka a 3 opakovací,
- každá vlajka reprezentuje i jednu často opakovanou větu, například vlajka O znamená „Muž přes palubu“,
- případně lze vlajky kombinovat do číselných kódů odkazujících do smluveného slovníku.

Vlajková abeceda – příklad

Slavný signál admirála Nelsona „England expects that every man will do his duty“ vyslaný v roce 1805 před bitvou u Trafalgaru vypadá ve vlajkové abecedě takto:



253	269	863	261	471	958	220	370	4	21	19	24
England	expects	that	every	man	will	do	his	D	U	T	Y

en.wikipedia.org/wiki/England_expects_that_every_man_will_do_his_duty

Definice základních pojmů

Definice

Konečná posloupnost x symbolů ze Σ se nazývá řetězec nad Σ.

Definice

Prázdná posloupnost se nazývá prázdný řetězec, značíme ji ε.

Definice

Délka řetězce x se značí $|x|$ a rovná se počtu výskytů symbolů v něm obsažených, $|\varepsilon| = 0$.

Definice

Množinu všech řetězců nad abecedou Σ bez prázdného řetězce budeme značit Σ^+. Množinu všech řetězců nad abecedou Σ, včetně prázdného řetězce, budeme značit Σ^*.

Definice základních pojmů (pokrač.)

Věta

Je zřejmé, že

$$\Sigma^* = \Sigma^+ \cup \{\varepsilon\}$$

Definice

Nechť x a y jsou libovolné řetězce, pak xy označuje konkatenaci (zřetězení) řetězců x a y.

Věta

Konkatenace dvou řetězců není obecně komutativní, tj.

$$\exists x, y \in \Sigma^+ : |xy| \neq |yx|$$

Dále je zřejmé, že

$$|xy| = |yx| = |x| + |y|$$

Definice základních pojmů (pokrač.)

Definice

Řetězec u se nazývá předponou (prefixem) řetězce w, jestliže existuje řetězec v (i prázdný) takový, že $w = uv$.

Definice

Řetězec v se nazývá příponou (sufixem) řetězce w, jestliže existuje řetězec u (i prázdný) takový, že $w = uv$.

Definice

Řetězec y se nazývá podřetězcem (faktorem) řetězce w, jestliže existují řetězce u a v (i prázdné) takové, že $w = uzv$.

Definice základních pojmů – příklady

- Abeceda $\Sigma = \{A, C, G, T\}$, řetězce $x = GCAGAGAG$ a $y = TAA$, $|x| = 8$ a $|y| = 3$.
- Konkatenace $xy = GCAGAGAGTAA$, $yx = TAAGCAGAGAG$, $|xy| = |yx| = |x| + |y| = 11$.
- Prefixem řetězce y jsou řetězce T, TA, TAA.
- Sufixem řetězce y jsou řetězce A, AA, TAA.
- Podřetězcem řetězce y jsou řetězce A, T, TA, AA, TAA.

Brute force algoritmus

```
1 size_t BruteForce(const string& Text, const string& Pattern, const
      size_t StartPosition)
2 {
3    for (size_t i = StartPosition; i < Text.length() - Pattern.length();
          i++)
4    {
5       size_t j = 0;
6       while (j < Pattern.length())
7       {
8          if (Text[i + j] != Pattern[j])
9             break;
10         j += 1;
11      }
12      if (j == Pattern.length())
13         return i;
14   }
15   return PatternNotFound;
16 }
```

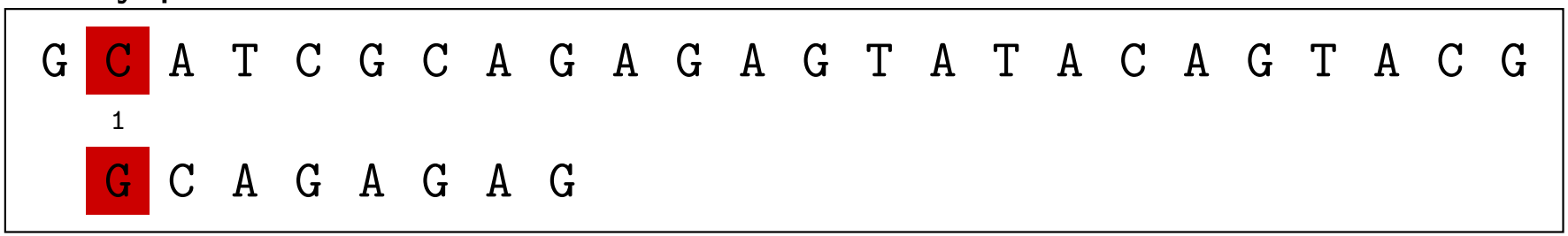
Brute force algoritmus

První pokus



Posun o 1 znak

Druhý pokus



Posun o 1 znak

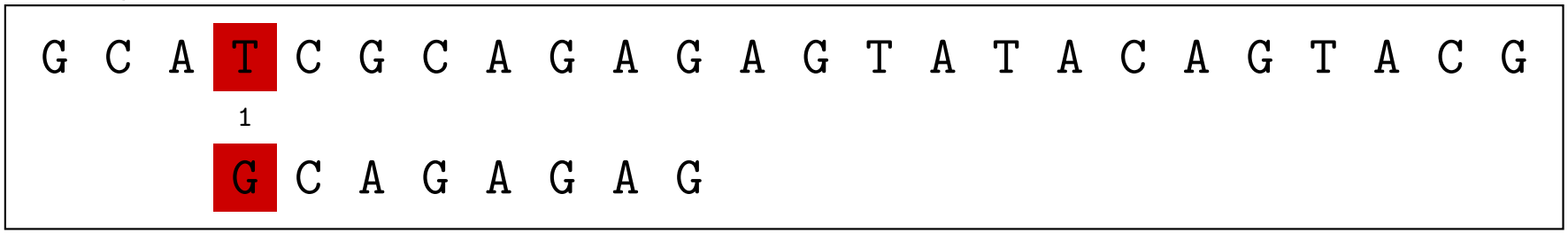
Brute force algoritmus (pokrač.)

Třetí pokus



Posun o 1 znak

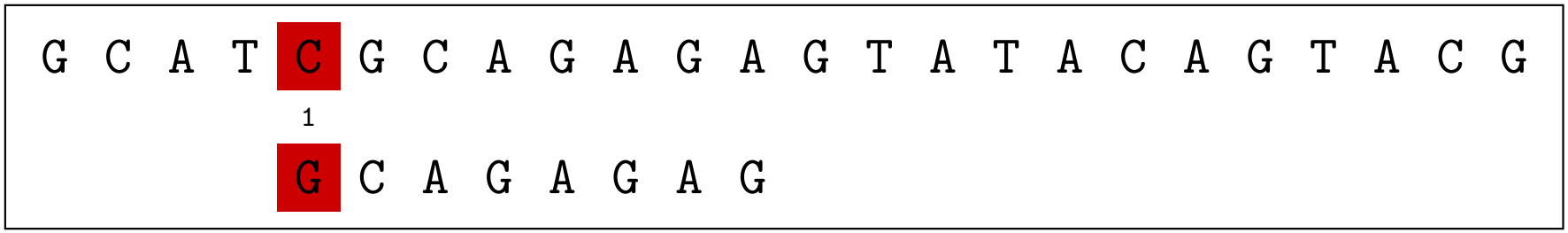
Čtvrtý pokus



Posun o 1 znak

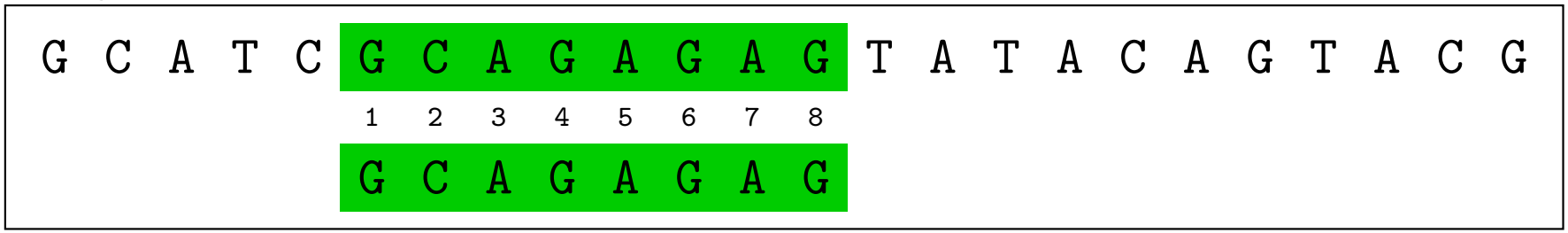
Brute force algoritmus (pokrač.)

Pátý pokus



Posun o 1 znak

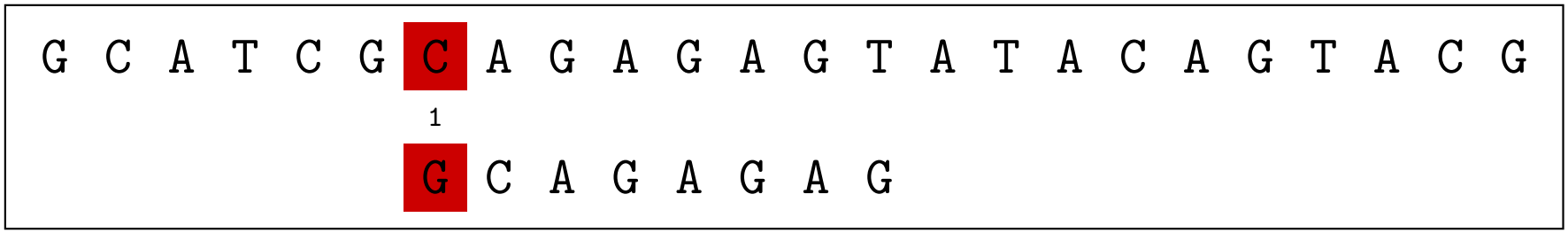
Šestý pokus



Posun o 1 znak

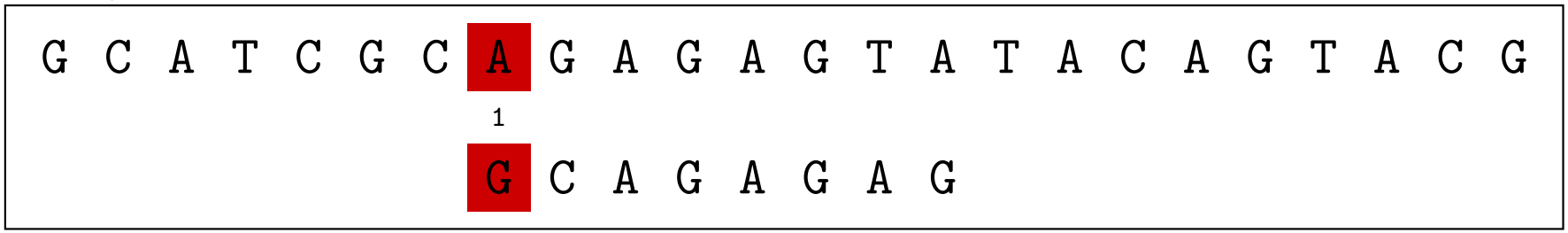
Brute force algoritmus (pokrač.)

Sedmý pokus



Posun o 1 znak

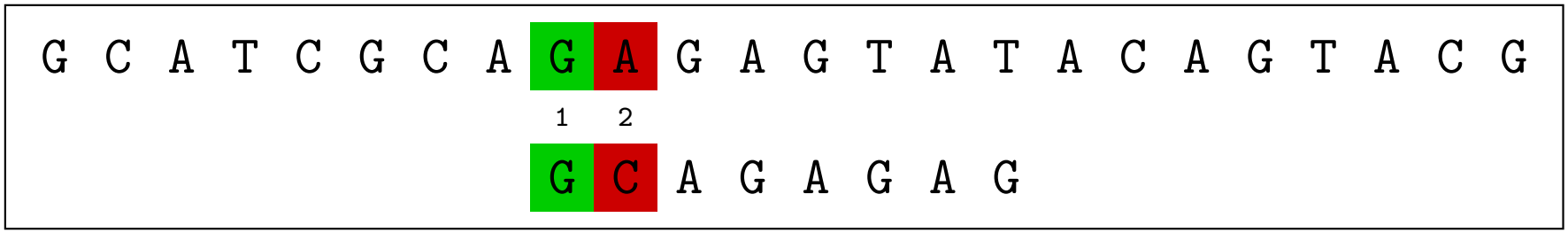
Osmý pokus



Posun o 1 znak

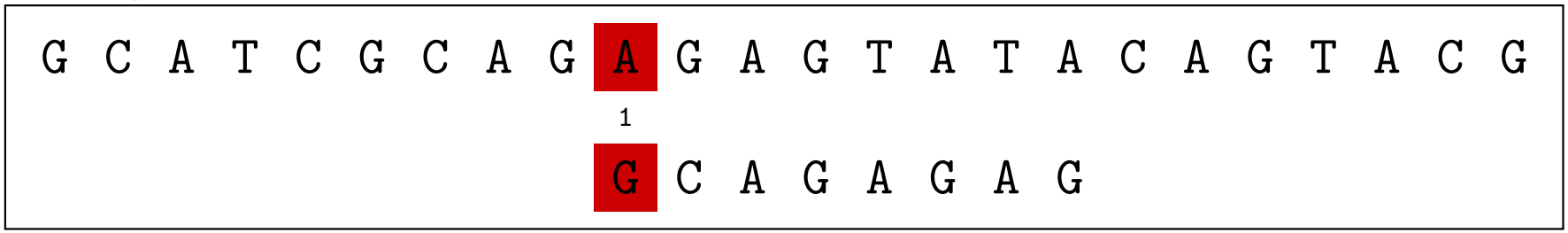
Brute force algoritmus (pokrač.)

Devátý pokus



Posun o 1 znak

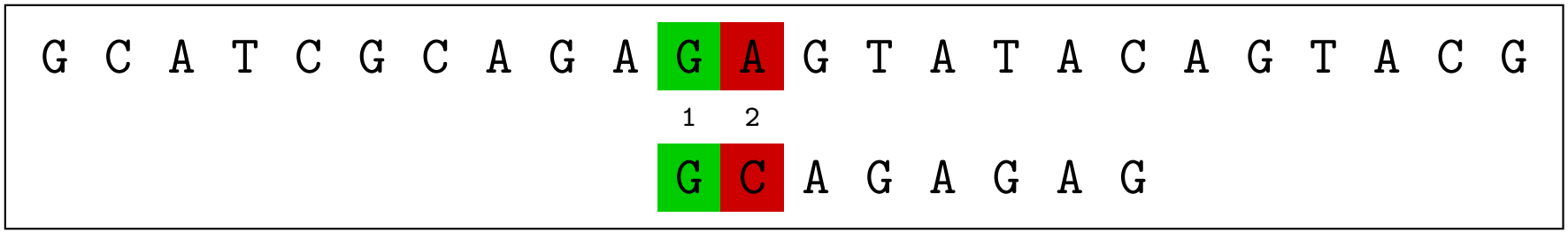
Desátý pokus



Posun o 1 znak

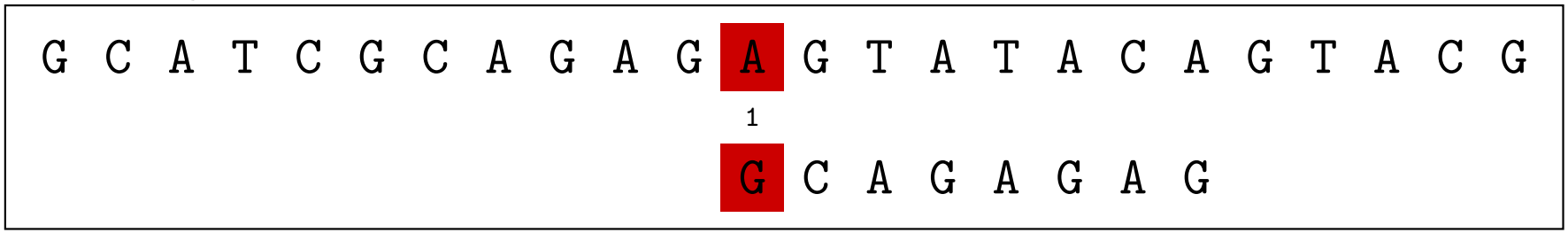
Brute force algoritmus (pokrač.)

Jedenáctý pokus



Posun o 1 znak

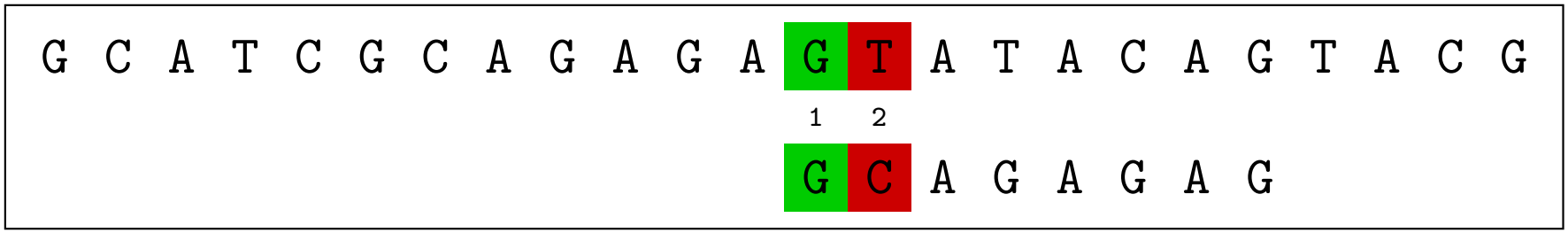
Dvanáctý pokus



Posun o 1 znak

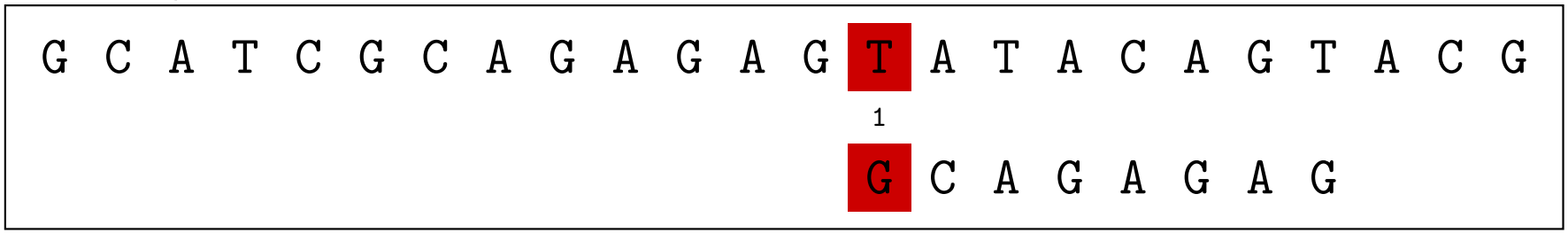
Brute force algoritmus (pokrač.)

Třináctý pokus



Posun o 1 znak

Čtrnáctý pokus



Posun o 1 znak

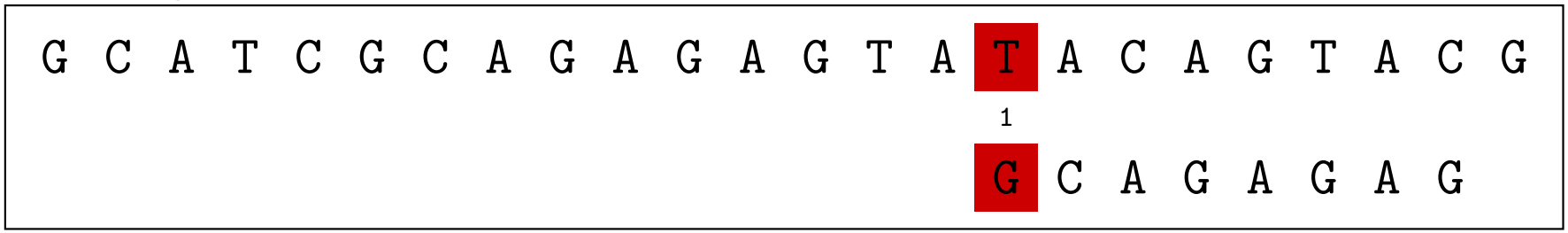
Brute force algoritmus (pokrač.)

Patnáctý pokus



Posun o 1 znak

Šesnáctý pokus



Posun o 1 znak

Brute force algoritmus (pokrač.)

Sedmnáctý pokus



Posun o 1 znak

Algoritmus provedl celkem 30 porovnání znaků.

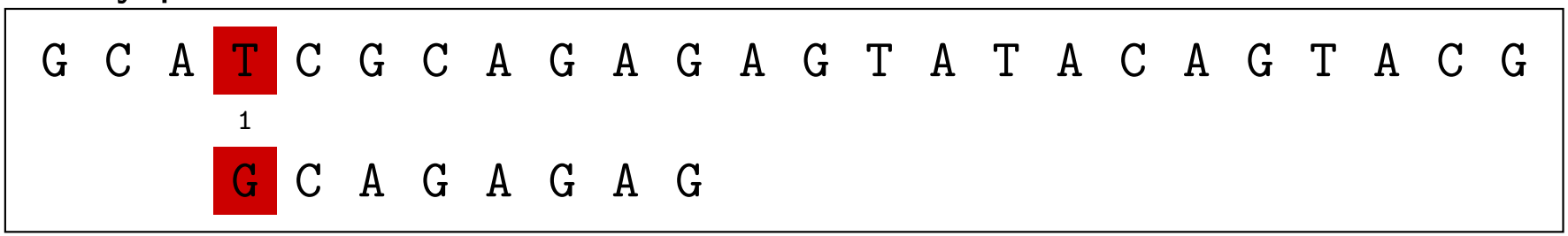
Morris-Prattův algoritmus

První pokus



Posun o $i - Next[i] = 3 - 0 = 3$ znaky

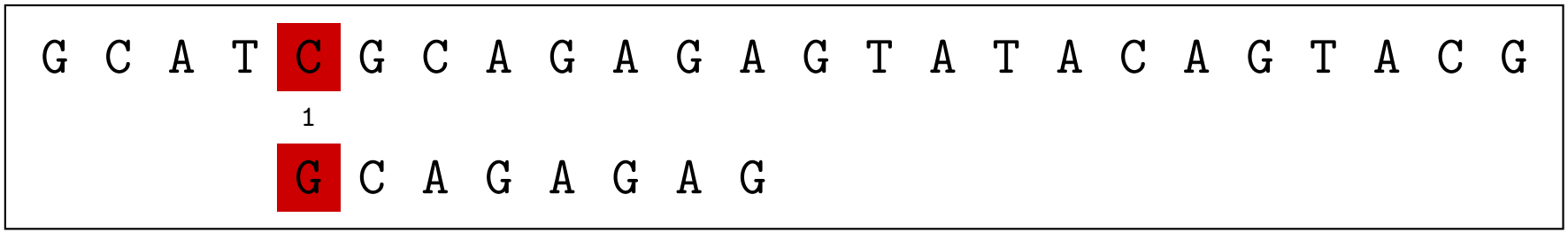
Druhý pokus



Posun o $i - Next[i] = 0 - (-1) = 1$znak

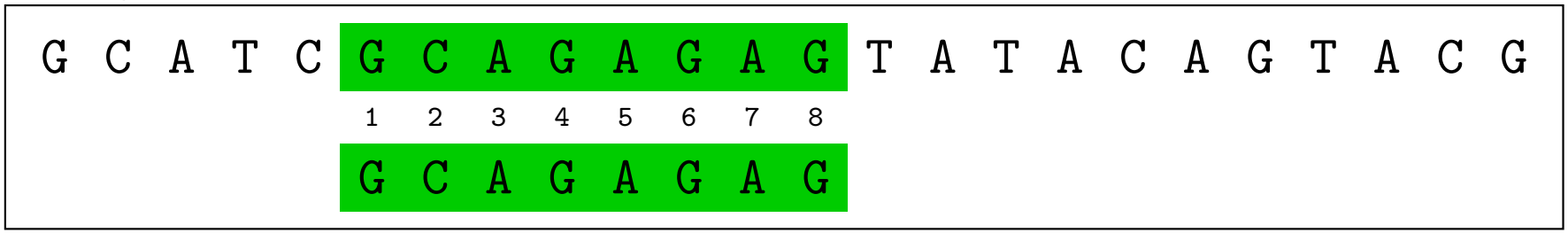
Morris-Prattův algoritmus (pokrač.)

Třetí pokus



Posun o $i - Next[i] = 0 - (-1) = 1$ znak

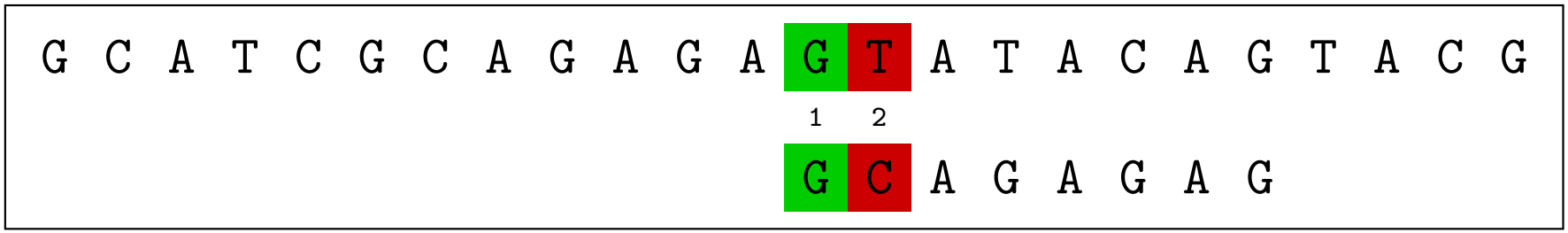
Čtvrtý pokus



Posun o $i - Next[i] = 8 - 1 = 7$ znaků

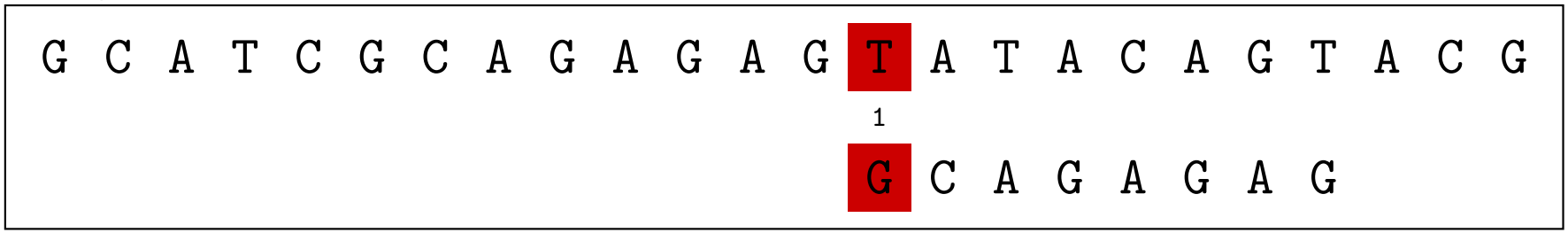
Morris-Prattův algoritmus (pokrač.)

Pátý pokus



Posun o $i - Next[i] = 1 - 0 = 1$ znak

Šestý pokus



Posun o $i - Next[i] = 0 - (-1) = 1$ znak

Morris-Prattův algoritmus (pokrač.)

Sedmý pokus



Posun o $i - Next[i] = 0 - (-1) = 1$ znak

Osmý pokus



Posun o $i - Next[i] = 0 - (-1) = 1$ znak

Morris-Prattův algoritmus (pokrač.)

Devátý pokus

G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
																1							
																G	C	A	G	A	G	A	G

Posun o $i - Next[i] = 0 - (-1) = 1$ znak

Algoritmus provedl celkem 19 porovnání znaků.

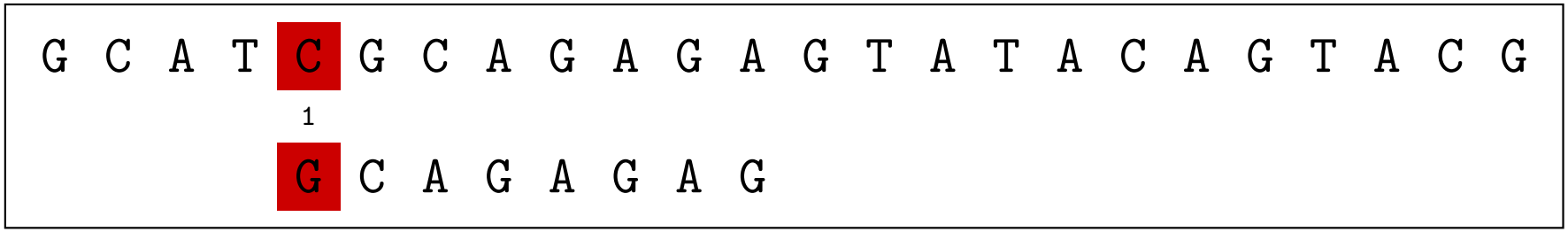
Knuth-Morris-Prattův algoritmus

První pokus



Posun o 4 znaky, tj. $i - Next[i] = 3 - (-1)$

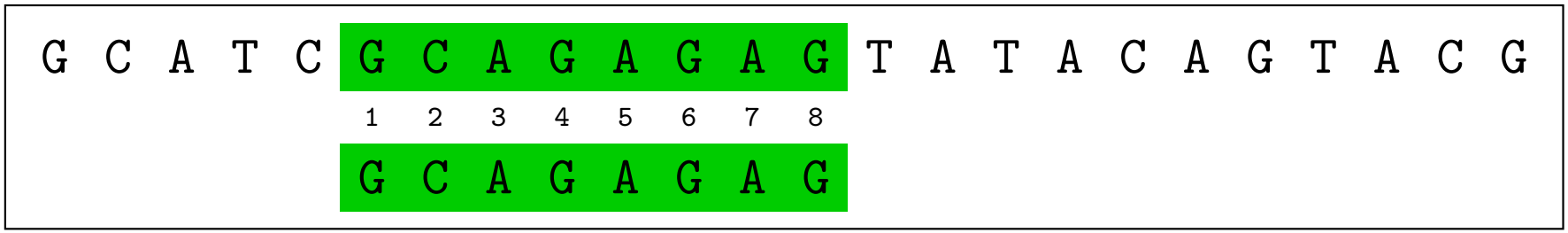
Druhý pokus



Posun o 1 znak, tj. $i - Next[i] = 0 - (-1)$

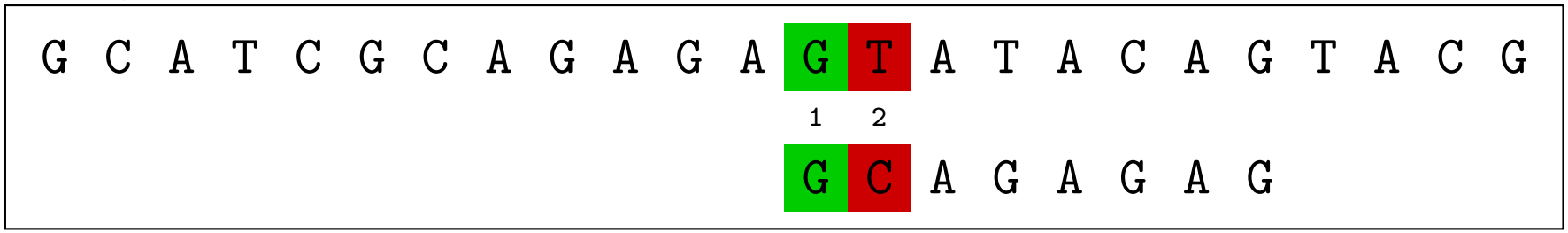
Knuth-Morris-Prattův algoritmus (pokrač.)

Třetí pokus



Posun o 7 znaků, tj. $i - Next[i] = 8 - 1$

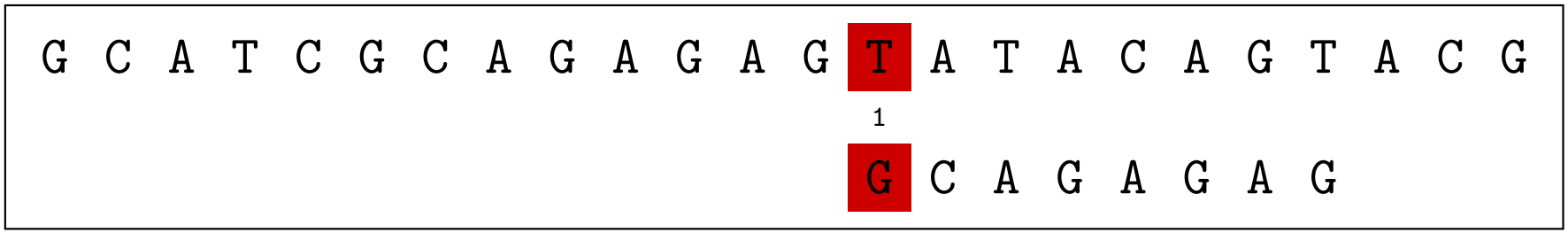
Čtvrtý pokus



Posun o 1 znak, tj. $i - Next[i] = 1 - 0$

Knuth-Morris-Prattův algoritmus (pokrač.)

Pátý pokus



Posun o 1 znak, tj. $i - Next[i] = 0 - (-1)$

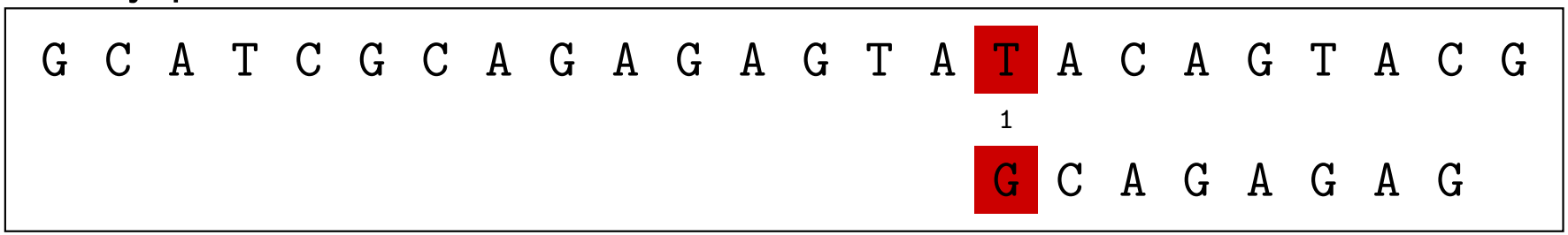
Šestý pokus



Posun o 1 znak, tj. $i - Next[i] = 0 - (-1)$

Knuth-Morris-Prattův algoritmus (pokrač.)

Sedmý pokus



Posun o 1 znak, tj. $i - Next[i] = 0 - (-1)$

Osmý pokus



Posun o 1 znak, tj. $i - Next[i] = 0 - (-1)$

Algoritmus provedl celkem 18 porovnání znaků.

KarpRabin

```
G C A T C G C A G A G A G T A T A C A G T A C G
G C A G A G A G
```

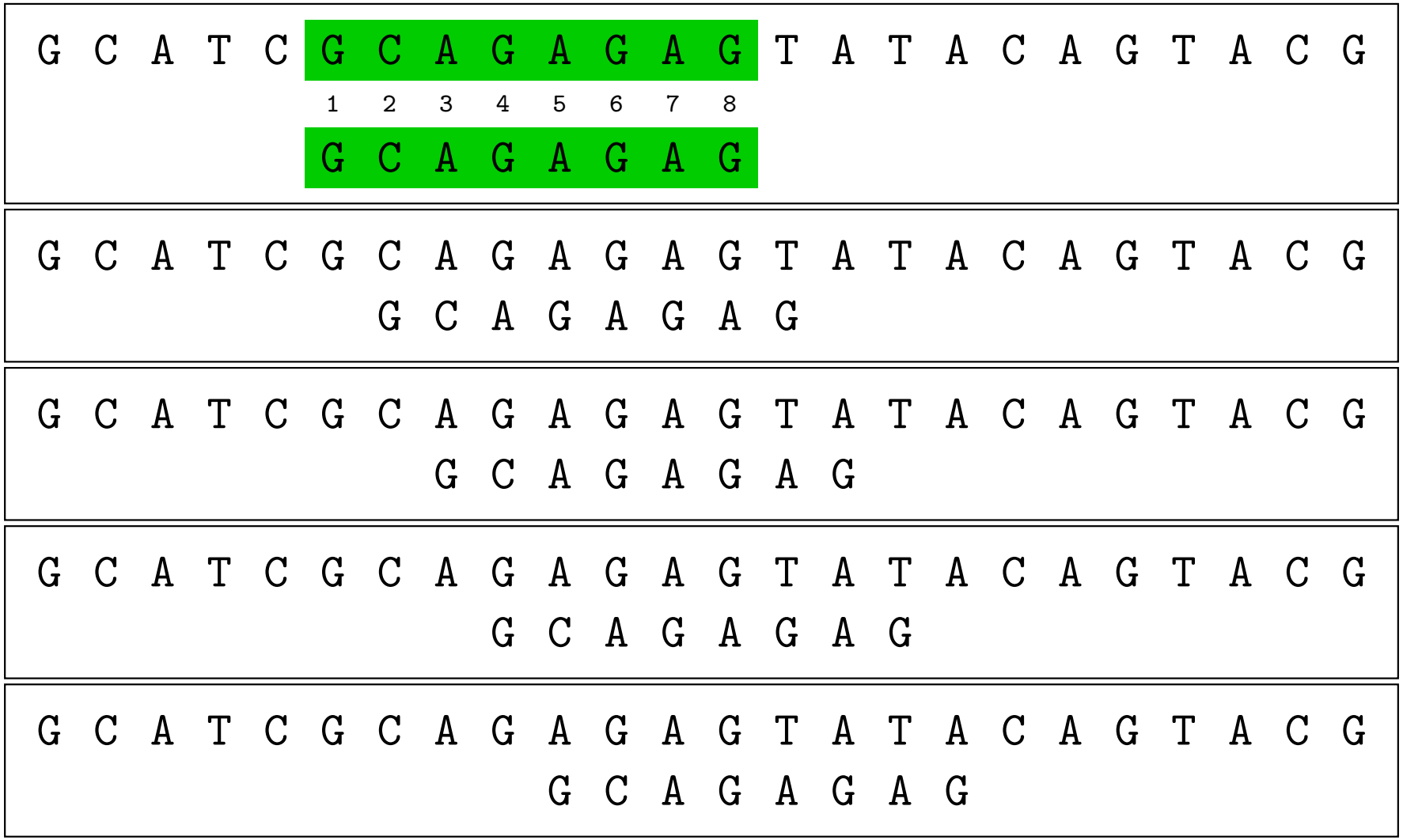
```
G C A T C G C A G A G A G T A T A C A G T A C G
  G C A G A G A G
```

```
G C A T C G C A G A G A G T A T A C A G T A C G
    G C A G A G A G
```

```
G C A T C G C A G A G A G T A T A C A G T A C G
      G C A G A G A G
```

```
G C A T C G C A G A G A G T A T A C A G T A C G
        G C A G A G A G
```

KarpRabin (pokrač.)



KarpRabin (pokrač.)

G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
										G	C	A	G	A	G	A	G						

G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
											G	C	A	G	A	G	A	G				

G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
												G	C	A	G	A	G	A	G				

G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
													G	C	A	G	A	G	A	G			

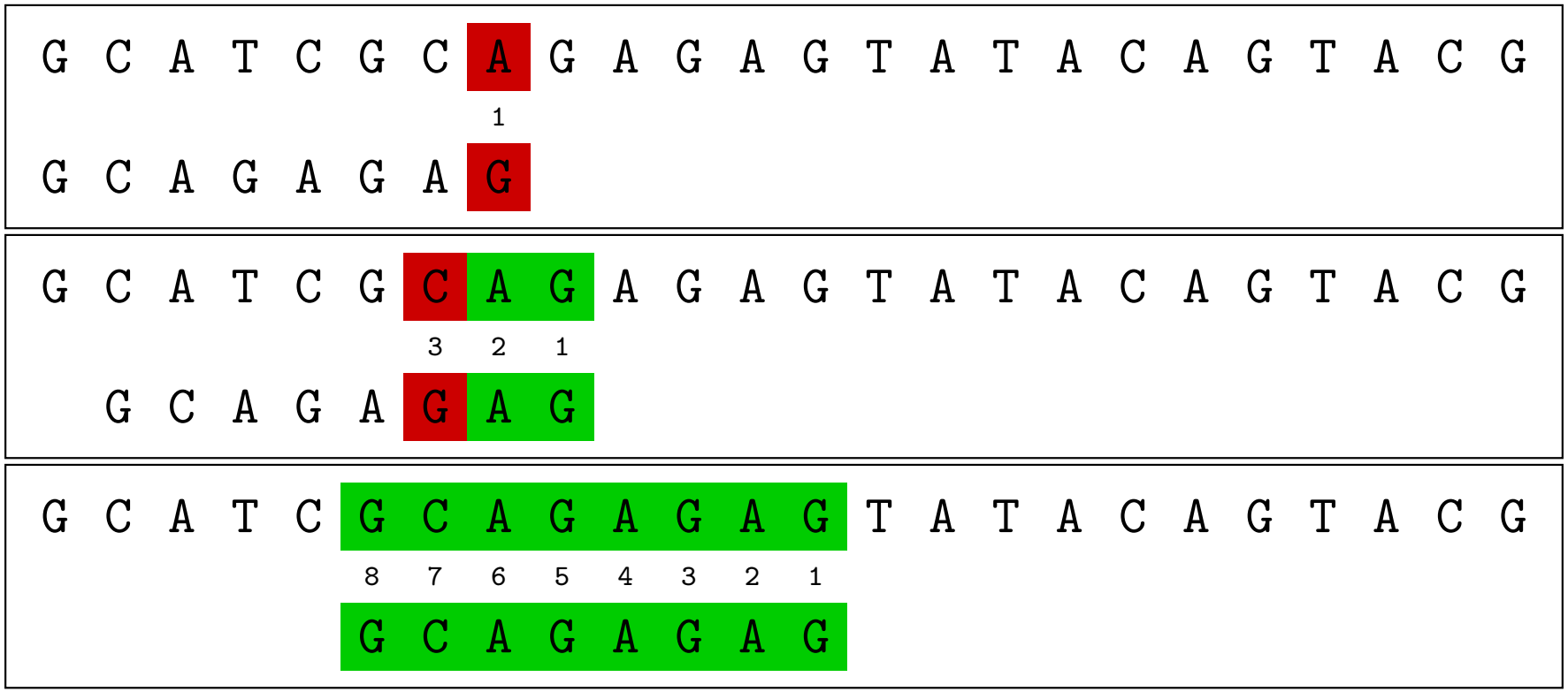
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
														G	C	A	G	A	G	A	G		

KarpRabin (pokrač.)

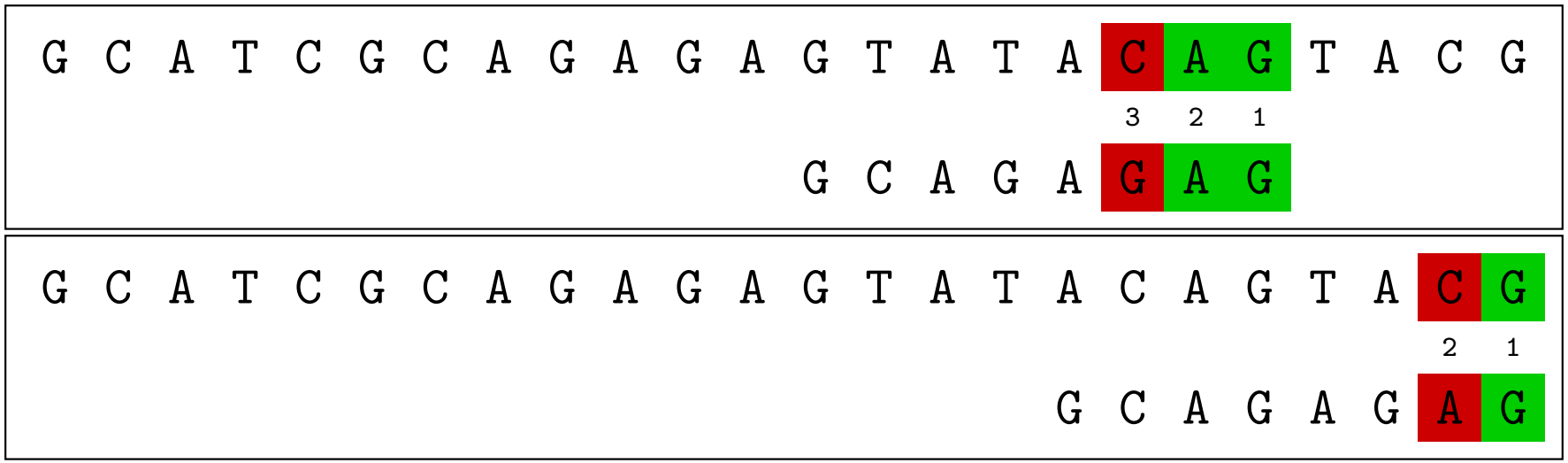
G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
															G	C	A	G	A	G	A	G	

G	C	A	T	C	G	C	A	G	A	G	A	G	T	A	T	A	C	A	G	T	A	C	G
																G	C	A	G	A	G	A	G

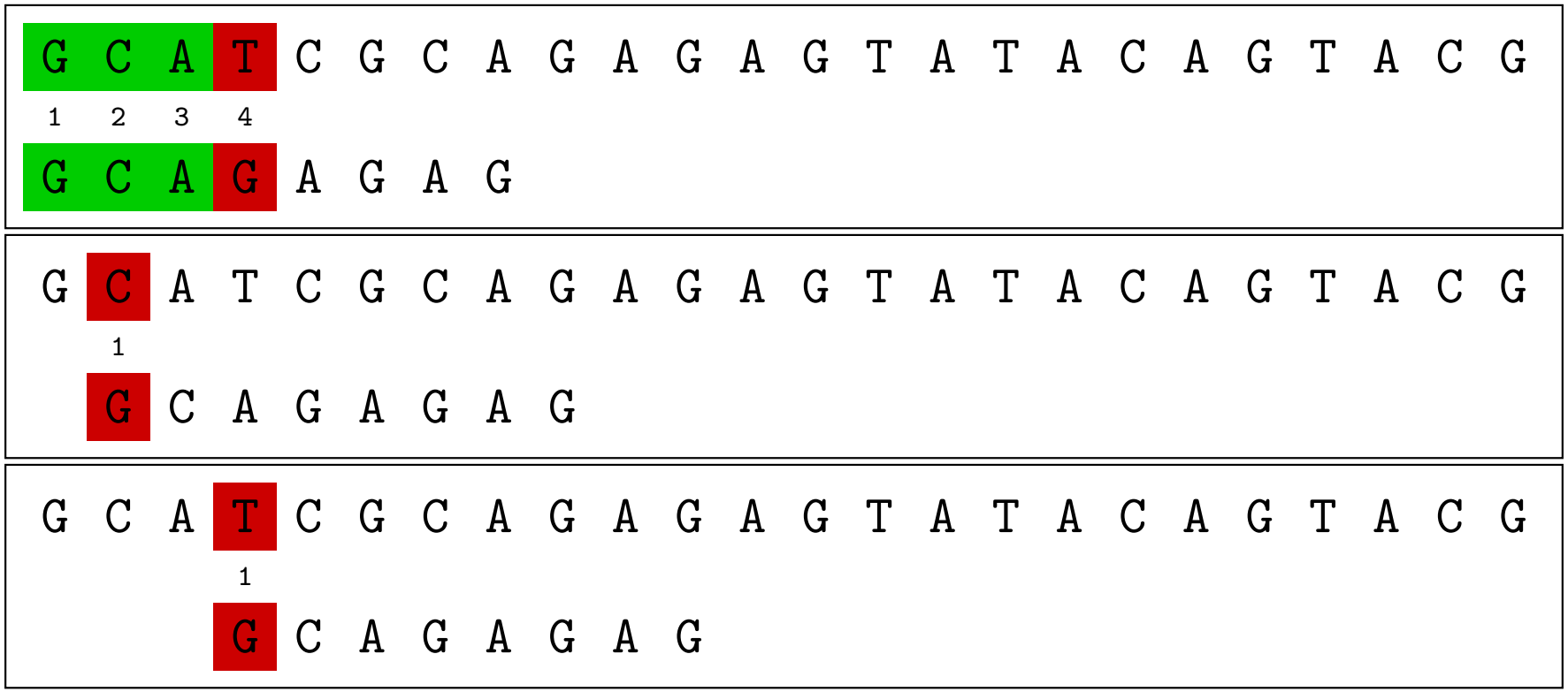
BoyerMoore


G C A T C G C A G A G A G T A T A C A G T A C G
1
G C A G A G A G
G C A T C G C A G A G A G T A T A C A G T A C G
3 2 1
G C A G A G A G
G C A T C G C A G A G A G T A T A C A G T A C G
8 7 6 5 4 3 2 1
G C A G A G A G

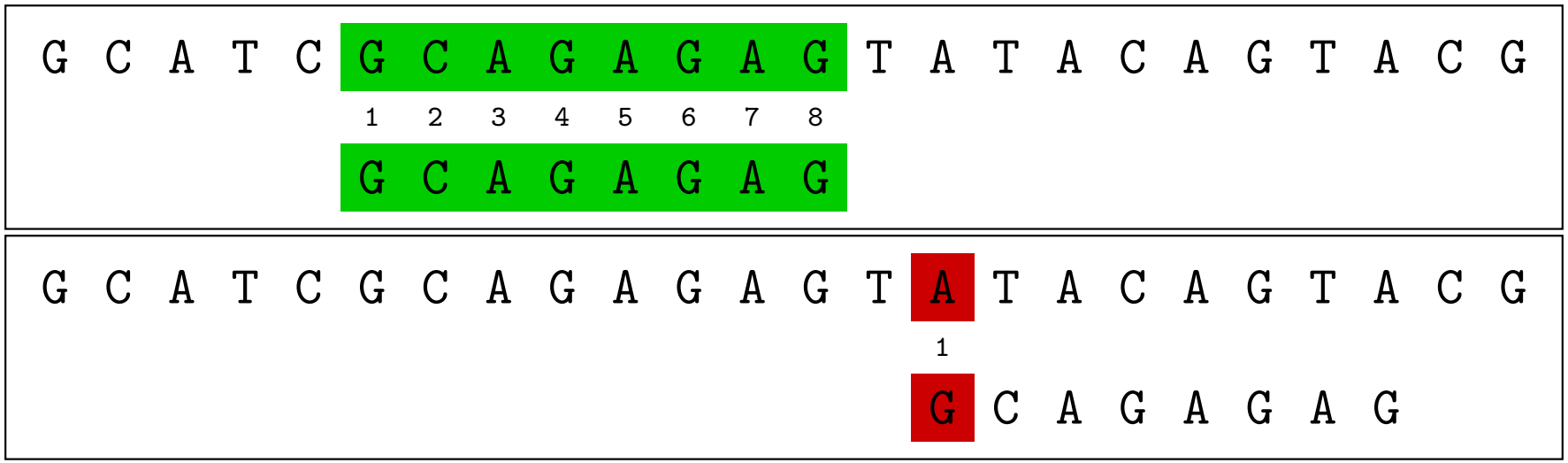
BoyerMoore (pokrač.)



QuickSearch



QuickSearch (pokrač.)



Kontrolní otázky

1. Definujte problém vyhledání vzorku v textu. Kde se lze s těmito problémy setkat?
2. Co je to předzpracování textu a vzorku?
3. Uveďte algoritmus, který pro svůj běh nepotřebuje předzpracovat ani text ani vzorek.
4. Uveďte příklad algoritmu (algoritmů), který předzpracovává vzorek ale nepředzpracovává prohledávaný text.
5. Uveďte příklad vyhledávací metody, která nepředzpracovává vzorek ale předzpracovává prohledávaný text.
6. Uveďte příklad vyhledávací metody, která předzpracovává vzorek i prohledávaný text.
7. Co je to abeceda? Označme ji například Σ.
8. Co je to řetězec nad abecedou Σ? Jak je definována délka řetězce?

Kontrolní otázky (pokrač.)

9. Máte dánu abecedu Σ a přirozená čísla m a n. Jak budou vypadat řetězce a^m a $a^m b^n$, kde $a, b \in \Sigma$?
10. Co je to prázdný řetězec? Jak jej značíme? Jaká je jeho délka?
11. Máte dánu abecedu Σ. Co označují množiny Σ^+ a Σ^*?
12. Co je to předpona (prefix) řetězce w?
13. Co je to přípona (sufix) řetězce w?
14. Co je to podřetězec (faktor) řetězce w?
15. Co je to hranice řetězce w?
16. Popište princip elementárního algoritmu (BruteForce Algorithm).
17. Jaká je časová složitost elementárního algoritmu? Jaké operace nás zajímají u výpočtu složitosti vyhledávacích algoritmů?
18. Jaká je očekávaná časová složitost elementárního algoritmu pro text a vzorek z přirozených jazyků?

Kontrolní otázky (pokrač.)

19. Elementární algoritmus provádí jednu činnost se vzorkem velice neefektivně. Tato neefektivita mu znemožňuje dosahovat menší časové složitosti při vyhledávání. O jako činnost je vzorkem jde a jak tuto neefektivitu řeší ostatní algoritmy? Například Knuth-Morris-Prattův?
20. Popište princip Shift-Or algoritmu. Jaké dvě bitové operace se tu používají?
21. Jaká je časová složitost Shift-Or algoritmu?
22. Popište princip Karp-Rabinova algoritmu. K čemu slouží u tohoto algoritmu hašovací funkce (Pozn. Vzpomeňte si na kapitolu o hašování a pohled na hašovací funkci jako na „negativní filtr“.)?
23. Co je to sousměrné a protisměrné vyhledávání?
24. Popište princip Boyer-Mooreova algoritmu. Soustřeďte se na využití protisměrného vyhledávání pro efektivní detekci (ne)výskytu vzorku v textu.

Kontrolní otázky (pokrač.)

25. Jaká je složitost Boyer-Mooreova algoritmu v nejhorším případě a jaká v nejlepším? Pro jakou kombinaci vzorku a textu dojde k nejhoršímu případu?

Děkuji za pozornost