✖ ALGORYTMY I STRUKTURY DANYCH – PRZESZUKIWANIE TEKSTÓW

- Wyszukiwanie wzorca w tekście – **algorytm „naiwny”**
  - **dane**: tekst T (np. tablica), wzorzec x (np. tablica)
    - długość T = n, długość x = m
    - symbole w T i x należą do pewnego alfabetu Σ
  - **wynik**: wszystkie wystąpienia wzorca w tekście

```
i ← 1
while(i ≤ n - m+1)
    if (x[1…m] = T[i…i+m-1])
     wypisz i
i ← i+1
```

- **złożoność**: O(m*n)

- Wyszukiwanie wzorca w tekście – **algorytm Boyera-Moore’a**

```
i← 1
while(i ≤  n - m+1)
    j←m
    while(x[j]=T[i+j-1])
        j←j-1
    if(j=0)
      wypisz i

i← i + s
```

- heurystyka niezgodności: $x[j] \neq T[i+j-1]$ (dla $1 \leq j \leq m$)
  - jeśli symbol nie występuje we wzorcu to s = j
  - jeśli symbol występuje na prawo od j to s=1
  - jeśli symbol występuje na lewo od j to s = j – ostatnie wystąpienie T[i+j-1] w x
    - tj. tak, by niepasujący symbol w T[i+j-1] pokrył się z identycznym w x

- s (przesunięcie) wyznaczane w oparciu o heurystykę niezgodności i heurystykę dobrego wyboru

- Wyszukiwanie wzorca w tekście – **algorytm Rabina-Karpa**
  - **założenie**: T ciąg „cyfr” należących do pewnego systemu o podstawie „r”

```
i← 1
while(i ≤  n - m+1)
    if(h_i(T_i) = h(x))
      if( x = T_i)  wypisz i
i← i + 1
```

  - h(x) – wartość funkcji haszującej dla wzorca (wartość dziesiętna odpowiadająca wzorcowi)
    - $h(x) = x[1]*r^{m-1} + T[2]*r^{m-2} + \ldots + T[m]$
  - $h_i(T_i)$ – wartość funkcji haszującej dla T[i…i+m-1]
    - $h_i(T_i) = T[i]*r^{m-1} + T[i+1]*r^{m-2} + \ldots + T[i+m-1] = (h_{i-1}(T_{i-1}) - T[i-1]*r^{m-1})*r + T[i+m]$