

## PARALELNÝ COUNTING SORT

Disclaimer: Nie som si istá svojou znalosťou WT prezentácie, tak som sa snažila jednotlivé kroky čo najzrozumiteľnejšie popísať slovne. Dúfam, že postačujúco. Okrem toho tu chýbajú horné celé časti, lebo mi ich OpenOffice nevie vypísať (resp. ho neviem prinútiť).

Vstupné pole  $A$  si rozdelím do úsekov dĺžky  $\log n$ . Na nich paralelne spočítam výskyty jednotlivých čísel v danom úseku. Tieto početnosti si ukladám do pomocného poľa  $B$  tak, že na  $i$ -tej pozícii sa nachádza výskyt čísla  $i \bmod \log n$  v príslušnom úseku (konkrétne  $i \operatorname{div} \log n + 1$ -vom).

Paralelne pracujem na úsekoch dĺžky  $\log n$  s poľom dĺžky  $n$ , teda  $T(n) = O(\log n)$ ,  $W(n) = O(n)$ . Zinicializovanie (vynulovanie) poľa  $B$  má  $T(n) = 1$ .

```
for i:=0 to n pardo
  B[i]:=0;
for i:=0 to  $\frac{n}{\log n}$  pardo
  for j:=i*log n to i*log n + log n
    inc(B[A[j]*i]);
```

Pomocou algoritmu 1.6.1 na výpočet sumy vypočítam výskyty v celom poli a to nasledovne: Každý proces spočíta výskyty jedného čísla  $c$ , teda  $c$ -ty proces spočíta políčka  $c + i * \log n$ , výsledok si uloží do pomocného poľa  $C$  na políčko  $C[c]$ .

Keďže políček je zjavne  $\frac{n}{\log n}$  a zložitosť spomínaného algoritmu je  $T(n) = \log n \rightarrow$

$$T\left(\frac{n}{\log n}\right) \leq O(\log n) \quad . \text{ Pre } W(n) = \log n * W\left(\frac{n}{\log n}\right) = \log n * O\left(\frac{n}{\log n}\right) = O(n).$$

```
for j:=0 to log n pardo
  for 1<= i<=  $\frac{n}{\log n}$  pardo
    set X(j,i):=B[i*log n + j];
  for k=1 to log( $\frac{n}{\log n}$ ) do
    for 1<=i<=  $\frac{\frac{n}{\log n}}{2^k}$  pardo
      set X(j,i):=X(j,2i-1)+X(j,2i);
    set C[j]:=X[1];
```

Teraz už mám pole  $C$ , dĺžky  $\log n$ , v ktorom sa na  $i$ -tej pozícii nachádza celkový výskyt čísla  $i$ .

Jedným sekvenčným prechodom pole upravím tak, že na  $i$ -tej pozícii budú všetky výskyty čísla  $i$  a čísel menších od neho (čiže najďalej možný výskyt  $i$  v konečnom poli).

Keďže  $C$  je dĺžky  $\log n$ ,  $T(n) = W(n) = \log n$ .

```
for i:=1 to log n - 1 do
  C[i]:=C[i] + C[i-1];
```

Aby sme v ďalšom predišli konfliktom v EREW pamäti, nakopírujem si do pomocného poľa  $D$  „odkiaľ“ sa má číslo vypísať.

Opät' pracujem sekvenčne s polom dĺžky  $\log n$ , čiže  $T(n) = W(n) = \log n$ .

```
for i:=0 to  $\log n - 1$  do
  if i=0 then D[i]:=0
  else D[i]:=C[i-1]+1;
```

Pomocou získaných informácií prepíšem pole  $A$  utriedenými hodnotami.

$T(n) = 1, W(n) = n$ .

```
for i:=0 to  $\log n - 1$  pardo
  for j:= D[i] to C[i] pardo
    A[j]:=i;
```