

Sirag – Descrierea soluției

Emanuela Cerchez

I.

Vom sorta descrescător vectorul a .

Considerăm primele p valori din sirul a și le vom scădea cu a_p (ceea ce ar însemna că vom plasa mărețele $1, 2, \dots, p$ de a_p ori).

Resortăm vectorul a și repetăm operația.

Complexitate: $O(k^2 \log k)$

II.

Observăm că după aplicarea unui pas, tabloul a este divizat în două subsecvențe descrescătoare.

Pentru sortare este suficient să interclasăm cele două secvențe.

Complexitatea va fi în acest caz $O(k^2)$.

III.

O altă soluție este să organizăm valorile a_1, a_2, \dots ca un *heap* și la fiecare pas să extragem de p ori maximum, reducem elementele și le reinserăm în *heap*.

Complexitate: $O(k p \log k)$.

IV

Complexitate: Sortare + $O(p \log s)$. (unde s = suma tuturor valorilor din a)

Sortăm vectorul a .

Calculăm în vectorul s șirul sumelor parțiale

$s[i] = a_i + a_{i+1} + \dots + a_{n-1}$

Vom căuta binar rezultatul în intervalul $[0, s[0]/p+2]$.

Funcția `good(x)` verifică dacă putem construi x secvențe de lungime p formate din valori distincte:

```
int good(long long x)
{for (i=0; i<p; i++)
    {if (x*(p - i) > s[i]) return -1;
     if (a[i] <= x) return i;}
  return p;
}
```

Funcția returnează valoarea -1 dacă nu este posibil, respectiv o valoare ≥ 0 (poziția) în caz contrar.

Lungimea șiragului se determină astfel:

$rez * p + rest$,

unde:

$rest = poz + s[poz] - rez * (p - poz)$