



Dorel - descrierea soluției

Autor: prof. Eugen Nodea

C. N. "Tudor Vladimirescu" Tg-Jiu

Se calculează liniar pentru fiecare poziție k ($1 \leq k \leq N$) din șir, poziția primului caracter mai mare sau egal cu $S[k]$ din stânga $stMax[i]$ și poziția primului caracter mai mic sau egal cu $S[k]$ din dreapta $drMin[j]$.

Folosind algoritmul lui Manacher (complexitate $O(N)$) vom preprocesa pentru șirul inițial, lungimea palindromului cu centrul în poziția i ($1 \leq i \leq N$). Pentru simplitate, ne vom referi doar la palindroamele de lungime impară, cazul cu lungime pară fiind asemănător (deoarece dacă avem un palindrom de lungime $2*i+1$, avem și palindroame de lungime $2*i-1, 2*i-3, \dots, 3, 1$, iar dacă avem palindrom de lungime $2*i$, avem și $2*i-2, 2*i-4, \dots, 4, 2$).

Pentru un query, vom numi X șirul/secvența care include poziția k și Y restul șirului concatenat.

Observăm că șirul X are o "anatomie" specială, și anume (în general) s-ar împărți în 5 zone:

i				k								j
\geq	<	<	<	=	>	>	>	>	>	>	>	\leq

Având preprocesate pentru șirul X lungimile palindroamelor, vom căuta binar lungimea palindromului maxim comun.

Verificare se face astfel:

- fie L lungimea palindromului pe care o căutam în acest moment.
- parcurgem X , iar când găsim un palindrom de lungimea L , folosim un algoritm de potrivire a șirurilor pentru a găsi acest palindrom în Y .
- dacă răspunsul este pozitiv, vom încerca să îmbunătățim lungimea maximă.

Pentru a evita cazurile când query-ul se face pe aceeași poziție, vom reține pozițiile calculate.

Complexitatea algoritmului: $O(Q*N*\log(\text{len}(X)))$.

Datorită anatomiei speciale ale lui X , în practică algoritmul se comportă mult mai bine.

Soluție cu hash-uri - prof. Ionel-Vasile Pit-Rada

Pentru fiecare întrebare se identifică cele două șiruri X și Y și se căuta binar lungimea celui mai lung palindrom-secvența comun. În cadrul căutării binare, pentru o lungime L fixată, se parcurge șirul X și se calculează pentru fiecare secvență de lungime L din X două hash-uri, $h1$ de la stânga spre dreapta și $g1$ de la dreapta spre stânga și dacă $h1 == g1$ atunci secvența este palindrom și se marchează într-un tablou la poziția $h1$. Apoi se parcurge șirul Y și se calculează asemănător cele două hash-uri $h1$ și $g1$ pentru fiecare secvență de lungime L din șirul Y și dacă $h1 == g1$ și $h1$ este deja marcat (la parcurgerea lui X) atunci se confirmă existența unui palindrom-secvența de lungime L comun celor două șiruri X și Y .



Complexitatea $O(Q \cdot n \cdot \log(n))$.

În practică, în funcție de implementare, algoritmul este mai lent.