**Dorel - descrierea soluției**

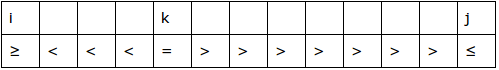
**Autor: prof. *Eugen Nodea***

**C. N. “Tudor Vladimirescu” Tg-Jiu**

Se calculează liniar pentru fiecare poziție **k** (1**≤**k**≤**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**≤**i**≤**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 si palindroame de lungime **2\*i-1**, **2\*i-3,**… , **3**, **1**, iar dacă avem palindrom de lungime **2\*i**, avem și **2\*i-2**, **2\*i-4,** …, **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 (in general) s-ar împărți in 5 zone:

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 in 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**.
* daca răspunsul este pozitiv, vom încerca să îmbunătățim lungimea maximă.

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

Complexitatea algoritmului: **O(Q\*N\*log(len(X))).**

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

**Solutie cu hash-uri - prof. Ionel-Vasile Pit-Rada**

Pentru fiecare întrebare se identifica cele doua șiruri **X** si **Y** si se căuta binar lungimea celui mai lung palindrom-secventa comun. In cadrul cautarii binare, pentru o lungime **L** fixata, se parcurge sirul **X** si se calculeaza pentru fiecare secventa de lungime **L** din **X** doua hash-uri , **h1** de la stanga spre dreapta si **g1** de la dreapta spre stanga si daca **h1==g1** atunci secventa este palindrom si se marcheaza intr-un tablou la pozitia **h1**. Apoi se parcurge sirul **Y** si se calculeaza asemanator cele doua hash-uri **h1** si **g1** pentru fiecare secventa de lungime **L** din sirul **Y** si daca **h1==g1** si **h1** este deja marcat (la parcurgerea lui **X**) atunci se confirma existenta unui palindrom-secventa de lungime **L** comun celor doua siruri **X** si **Y**.

Complexitatea **O(Q\*n\*log(n)).**

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