



## Problema 2 – cerc

Autor: Rodica Pinte

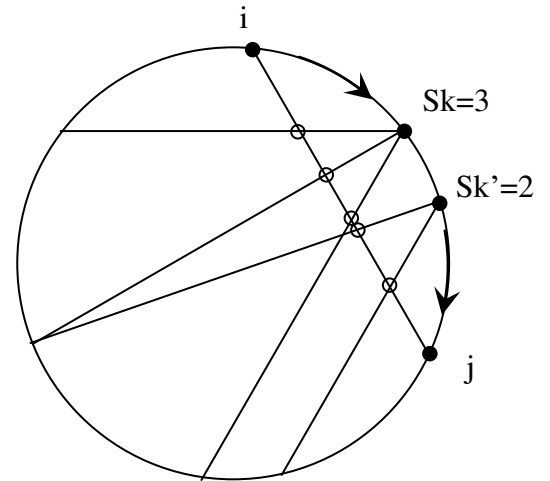
Soluție

Se rețin segmentele astfel:  $[i, j]$  dacă de la  $i$  spre  $j$  există mai puține puncte intermediare pe cerc sau  $[j, i]$  în caz contrar.

Se sortează liniar segmentele după lungimea lor relativă (lungimea relativă = numărul minim de puncte intermediare care se găsesc pe cerc între capetele segmentului).

Pentru fiecare punct  $i$  de pe cerc,  $i \in \{1 \dots N\}$ , se memorează numărul de segmente  $S_i$  care au ca extremitate punctul  $i$ .

În ordinea lungimilor relative, pentru fiecare segment  $[i, j]$ , se însumează valorile  $S_k$  pentru toate punctele intermediare de pe cerc (de la  $i$  către  $j$ ) și se elimină segmentul respectiv actualizând valorile  $S_i$  și  $S_j$ .



```
const nmax=500;mmax=nmax*(nmax-3)div 2;
      fi='cerc.in';fo='cerc.out';
type segm=record p1,p2,lung:integer end;
var n,j:integer;m,i:longint;
    p,l:array[0..nmax]of longint;
    ind:array[1..mmax]of longint;
    per:array[1..mmax]of segm;
    s:segm;inters:longint;
    f:text;
```

```
procedure calc(var s:segm);
var aux:integer;
begin
  if s.p2-s.p1<=n+s.p1-s.p2 then
    s.lung:=s.p2-s.p1-1
  else begin
    s.lung:=n+s.p1-s.p2-1;
    aux:=s.p1;s.p1:=s.p2;s.p2:=aux
  end
end;
```

```
procedure citire;
var i:longint;
begin
  assign(f,fi);reset(f);readln(f,n,m);
  i:=0;
  while not seekeof(f) do begin
    readln(f,s.p1,s.p2);
    calc(s);
    if s.lung=0 then continue;
    inc(p[s.p1]);inc(p[s.p2]);
    inc(l[s.lung]);i:=i+1;per[i]:=s
  end;
close(f);
end;
```

```
procedure sort;
var i,j:longint;
begin
  for i:=2 to n do l[i]:=l[i]+l[i-1];
  for j:=1 to m do begin
    i:=per[j].lung;
    ind[l[i]]:=j;
    dec(l[i])
  end
end;

begin
  citire;
  sort;
  inters:=0;
  for i:=1 to m do begin
    s:=per[ind[i]];
    j:=s.p1 mod n+1;
    while j<>s.p2 do begin
      inters:=(inters+p[j])mod 1000000;
      j:=j mod n+1
    end;
    dec(p[s.p1]);dec(p[s.p2])
  end;
  assign(f,fo);rewrite(f);
  writeln(f,inters);
  close(f)
end.
```