**Problema cerc - autor prof.Carmen Mincă** (Liceul "Ion Neculce", Bucureşti)

---

## 1. Descrierea soluţiei - prof.Constantin Gălăţan (sursa `cerct.cpp – 100p`)

Cercurile situate cu centrele pe o aceeaşi dreaptă determină un şir de intervale închise. Se reţin coordonatele centrelor cercurilor în două şiruri `x` şi `y` de numere reale. Se sortează cercurile crescător după unghiul pe care îl fac cu axa **Ox**, iar pentru unghiuri egale, se sortează crescător după capetele din dreapta ale intervalelor. Apoi, pentru fiecare dreaptă distinctă (corespunzătoare unei secvenţe consecutive în tablourile `x` şi `y`), se aplică o metodă de tip greedy asemănătoare problemei spectacolelor. Complexitate `O(n *log n)`.

## 2. Descrierea soluţiei - drd.Pătcaş Csaba (sursa `cerc.pas– 100p`)

Două cercuri vor avea centrul pe aceaşi dreaptă, dacă tangenta unghiului faţă de Ox a dreptelor determinate de origine şi centrul cercurilor este egală. Pentru a evita problemele cu precizia numerelor reale, vom reţine tangentele în forma unor fracţii ireducibile. Determinarea dreptelor se poate face în timp O(m*n).

Dacă pentru fiecare dreaptă considerăm numai punctele de intersectie cu cercurile a căror centru se află pe dreaptă, problema devine echivalentă cu determinarea unui set independent maximal de intervale, numit şi problema spectacolelor: sortăm intervalele crescător după capătul din dreapta, după care printr-o parcurgere a şirului sortat determinăm soluţia. Complexitatea acestui pas depinde de complexitatea sortării, deci se poate face cu uşurinţă în `O(n*log n)` folosind un algoritm clasic de sortare, cum ar fi QuickSort. Deoarece acest pas se repetă pentru fiecare dreaptă în parte, complexitatea totală va fi `O(m*n*log n)`

## 3. Descrierea soluţiei - prof.Carmen Mincă

(sursele `cerc_c.cpp, cerc_p.pas, cerc100.cpp cerc_g.cpp - 100p`)

Se pot construi soluţii cu complexitate `O(n*n)` sau `O(n*log n)`, aplicând metoda programării dinamice sau metoda Greedy.

O soluţie obţinută prin aplicarea metoda programării dinamice, metoda înainte, poate utiliza un tablou `c` unidimensional de tip înregistrare declarat astfel:

```
struct cerc  {int x,y,r,d,e;} c[500];
```

pentru a memora în `c[k]` (`k=1,2,...,n`) toate informaţiile referitoare la cercul `k` (coordonatele `x` şi `y`, raza `r`). Cîmpul `d` al structurii va memora numărul dreptei, dintre cele `m`, care trece prin centru cercului `k`. Câmpul `e` va memora numărul cercurilor cu centrele pe dreapta `d`, exterioare două câte două, şi cu abscisa centrului mai mare decât abscisa cercului `k`.

Se poate utiliza un tablou unidimensional de tip întreg `v` în care `v[k]` va memora numărul cercurilor ale căror centre sunt situate pe dreapta `k` şi sunt exterioare două câte două. Numărul valorilor distincte din tabloul `v` este egal cu numărul `m` de drepte distincte cerute.

Se sortează tabloul `c`, crescător după valorile câmpului `x` (abscisele centrelor cercurilor) şi după tangenta unghiului dintre axa `Ox` şi dreapta care trece prin `O` şi centrele cercurilor prin Qsort.

Plecând de la primul cerc, stabilim care dintre cercurile memorate în `c` la dreapta celui curent, au centru situat pe dreapta care trece prin centrul primului cerc. Două cercuri, de raze `r1` respectiv `r2`, au centrele (`x1,y1`) şi (`x2,y2`) situate pe aceeaşi dreaptă `d` care trece prin punctul **O** dacă este satisfăcută relaţia:

$$y2 = \frac{y1}{x1} \cdot x2 \Leftrightarrow y2 \cdot x1 = y1 \cdot x2$$

Pentru toate aceste cercuri se va memora în câmpul `d` acelaşi număr care reprezintă numărul dreptei curente (pentru primul cerc, numărul este `1`). Se procedează analog pentru toate cercurile care nu au iniţializat câmpul `d`. La final, fiecare `c[k].d` va memora un număr natural nenul cel mult egal cu `m`.

Două cercuri, cu centrele (x1,y1) şi (x2,y2), şi razele r1, respectiv r2, sunt exterioare dacă este satisfăcută relaţia: dist((x1,y1),(x2,y2))>r1+r2 <==> $(x1-x2)^2+(y1-y2)^2>(r1+r2)^2$

Începând cu penultimul cerc, se determină pentru fiecare cerc i numărul maxim al cercurilor j (j>i) cu centrele pe dreapta d care conţine şi centrul cercului i şi sunt exterioare două câte două. Acest număr se va memora în c[i].e (iniţial c[i].e=1, i=1,2,...,n):

Cel mai mare număr memorat în câmpul e al tabloului c reprezintă numărul q maxim de cercuri exterioare două câte două ale căror centre aparţin aceleiaşi drepte care trece prin punctul O. Numărul componentelor tabloului c care memorează în câmpul e valoarea q reprezintă numărul p al dreptelor care conţin centrele a câte q cercuri exterioare două câte două. (sursele `cerc_c.cpp` şi `cerc_p.pas` obţin 100p).

O altă soluţie se poate obţine reducând cercurile la intervalele închise rezultate prin intersectarea cercurilor cu dreptele pe care se află centrele lor, obţinându-se pentru fiecare dreaptă un şir de intervale închise. Problema se reduce la a determina numărul maxim de intervale disjuncte două câte două pentru fiecare dreaptă. Cu o adordare gen "problema subşirului maximal", se obţine punctajul maxim (sursa `cerc100.cpp`). O determinare prin Greedy, abordare gen "problema spectacolelor", obţine punctajul maxim (sursa `cerc_g.cpp`).