Az itt található `sorba.c` program egy valószínűségi iterált függvényrendszer invariáns mértékének spektrumának közelítő számolását szolgálja. A probléma Simon Károlytól és Hernáth Szabolcstól származik.

További információt is Simon Károly tud. (`http://www.math.bme.hu/~simonk`)

A szóban forgó IFS a $[0;1]$ intervallumon él, és három függvényből áll:

$$f_1(x) = \frac{x}{3}; \qquad f_2(x) = \frac{x}{3} + \frac{2}{9}; \qquad f_3(x) = \frac{x}{3} + \frac{2}{3}.$$

Az iteráció során mindhárom függvény súlya azonos. Lényeges elem, hogy az 1-es és 2-es függvény értékkészlete átfed, ráadásul pont $\frac{1}{3}$ részben, így a dinamika kombinatorikusan nyomonkövethető.

A program közvetlen célja, hogy valamely (mindegy, melyik) pontból kiindulva, és n iterációt végrehajtva, megkeresse a különböző elérhető pontok súlyát. Pontosabban azt, hogy n lépés után milyen súlyú pontból hány van. Mivel minden súly $\frac{1}{3^n}$-nek többszöröse, a program ezt kezeli egységnek, és egész számokkal számol.

Ennek megfelelően a program bemenete egyetlen n szám, kimenete pedig

- maga az n, vagyis az iterációk száma
- az n lépés során elérhető legnagyobb súly (N), ami egyben a kimeneti tömb mérete is
- a keletkező pontok összsúlya, ami persze $S = 3^n$ kell, hogy legyen
- a ténylegesen elért legnagyobb súly, ami persze N kell, hogy legyen
- végül egy hosszú felsorolás, ahol az i függvényében megkapjuk az i súlyú pontok számát, a_i-t ($i = 1 \dots N$).

Ezekből az adatokból számolható a mérték spektrumának egy közelítése a következő képlettel:

$$\tau(q) = \frac{\ln \sum_{i=1}^{N} \left(\frac{i}{S}\right)^q a_i}{(1-n)\ln 3 - \ln 2}.$$

Az érdekes mennyiség ez a $\tau(q)$ és az ő Legendre-transzformáltja, különös tekintettel a határokon való viselkedésre. Ezért volt cél, hogy az iterációt minél nagyobb n-re le lehessen futtatni.

A program erőssége, hogy kihasználva a rendszer kombinatorikáját, a benne szereplő legbolyolultabb művelet az összeadás. Tujadonképpen az n hosszú $\{0,1,2\}$ sorozatokat járja végig backtracking módszerrel, úgy, hogy a $(0,2)$ és az $(1,0)$ szavak azonosítva vannak. Közben számolja, hogy ki hányféleképpen található meg (vagyis mennyi a pont súlya), és a megfelelő súlyhoz húz egy strigulát.

A futási idő persze exponenciális n-ben, de így is simán el lehetett menni $n = 25$-ig, ez kb. két óra alatt lefutott az intézeti szerveren.

A program mellett megtalálhatók a futási eredmények sok n-re. A kimenetekből a $\tau(q)$ számolása és ábrázolása, valamint Legendre-transzformálása az olvasó feladata.