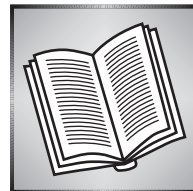


Monte-Carlo-módszerek 1.



Monte-Carlo híres a kaszinóiról és az ott folyó szerencsejátékról. A szerencsejátékok kapcsolata a matematikával a XVII. században Pascal és Fermat munkásságával kezdődött. A szerencsejátékok, és általában a véletlen jelenségek több érdekes és nehezen megoldható matematikai problémát eredményeztek. A megoldásukkal olyan kérdésekre kaptak választ, melyek régóta foglalkoztatták a játékosokat. A kedvelt játékokról megállapították, hogy melyik fél számára mennyire „nyerő”. Természetesen az is nyilvánvaló volt, hogy a kiszámított valószínűségeket egyetlen játékra nincs értelme alkalmazni, hanem sok játék eredményét kell följegyezni és összegezni. A játékok nem vesztek varázsukból, ugyanakkor igazi kihívás volt a matematikusoknak egy-egy valószínűség kiszámítása.

A XX. század közepén a számítógépek megjelenésével a szerencsejáték és a matematika is új lehetőségeket nyert. A kiszámított valószínűségeket „ellenőrizni” is lehetett nagyszámú kísérlettel. Csak meg kellett tanítani a számítógépet kockát dobni: a gép N -szer elvégez egy kísérletet, és minden kísérletben véletlenszerűen választ a lehetőségek közül. Az így nyert relatív gyakoriságok és a matematikai úton számított valószínűségek összevethetők voltak.

A Monte-Carlo-módszerek egy speciális problémamegoldási lehetőséget jelentenek, melyeket sikerrel alkalmaztak a matematika, fizika valamint az informatika területén. Az alap gondolat az előbbi „ellenőrzés” fordítottja. Mit tehetünk akkor, ha a vizsgált jelenség bizonyos értékei nem kiszámíthatók, tehát nincs zárt, könnyen kiértékelhető formula az eredmény meghatározására? Az egyik út már ismert volt a számítógépek előtt: numerikus módszerekkel meg lehetett határozni olyan számítási lépéseket, melyek megfelelő pontossággal megközelítik a keresett mennyiségeket. A numerikus módszerek alkalmazásában is nagy segítség volt a számítógép. A Monte-Carlo-módszer mást javasol: szimuláljuk sokszor a problémához kapcsolható kísérletet, és a kapott eredmények alapján adjunk közelítést a keresett ismeretlen értékekre.

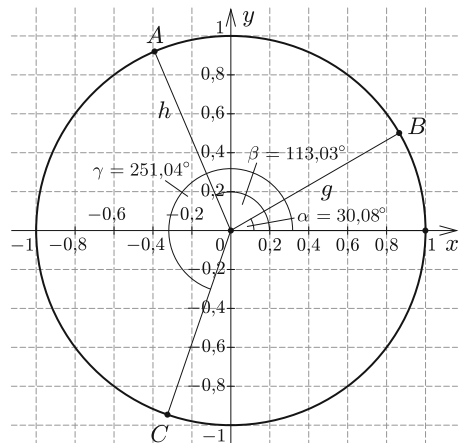
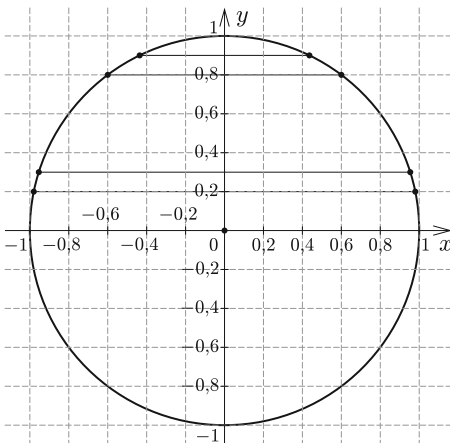
Lássuk először egy klasszikus valószínűségszámítás feladatnál a módszer működését: a 2008. márciusi számunkban megjelent **B. 4080.** feladat: A körvonalon egymástól függetlenül véletlenszerűen kiválasztunk 3 pontot. Mi annak a valószínűsége, hogy az általuk meghatározott háromszög hegyesszögű?

A feladat több szép megoldása olvasható honlapunkon, ezért nem volna szükségünk a Monte-Carlo-módszerre, hogy az eredményt közelítsük, mégis tanulságos a példa. A kísérletben véletlenszerűen elhelyezünk pontokat egy egység sugarú körön, majd meghatározzuk a kapott háromszög szögeit, és megállapítjuk, hogy hegyesszögű-e. A kísérletet N -szer elvégezve megszámláljuk a hegyesszögű háromszögeket. Ha ezek száma k , akkor a kísérletsorozatban a hegyesszögű háromszögek előfordulásának k/N relatív gyakoriságát kapjuk, ami elég nagy N esetén megfele-

lően pontos lesz. Kérdés ugyanakkor, hogy pl. két tizedesjegy pontossághoz milyen N értéket érdemes megadni, de ezt hagyjuk későbbre, most foglalkozunk a kísérletsorozatot megvalósító programmal.

Próbáljuk meg elkészíteni a számítógépes programot. Legyen egy egységkör a koordináta-rendszer kezdőpontjában, és $A(ax, ay)$, $B(bx, by)$ és $C(cx, cy)$ a véletlenszerűen választott csúcsok. A legtöbb programozási nyelvben megtalálhatók véletlenszámok, pl. egy valós értéket kaphatunk a $[0; 1[$ intervallumból (jelöljük ezt RND -vel). Azt a kérdést, hogy a számítógép miként állít elő véletlen értéket, hagyjuk későbbre. Amit biztosan tudnunk kell RND -ről, hogy egy futó program N egymást követő RND értékét ábrázolva az egység intervallumon, azok nagyjából egyenletes sűrűséggel helyezkednek el. Azaz minél nagyobb a pontok N száma, annál jobban teljesül, hogy az intervallum bármely szakaszának hossza, és a rajta megtalálható pontok száma között egyenes arányosság áll fenn. Első ötletünk, hogy RND értékét 2-vel megszorozva és 1-gyel csökkentve máris adódik mindhárom csúcs y koordinátája. Az x koordináták számításához használjuk a Pitagorasz-tételt, illetve egy-egy másik véletlenszámot: ha ez utóbbi 0,5-nél kisebb, akkor a pozitív, egyébként a negatív x koordinátát választjuk. Így véletlenszerűen kijelöltünk három pontot a köríven, már csak a szögeket kell kiszámolni. A kísérletet sokszor elvégezve egy relatív gyakorisággal tudjuk közelíteni a keresett értéket.

Gondoljuk meg, hogy helyesen jártunk-e el. Tanulmányainkból tudjuk, hogy a valószínűség klasszikus kiszámítási módja, a kedvező és összes esetek hányadosa csak akkor adja a valószínűséget, ha a kísérletben szereplő elemi események mind egyenlő valószínűséggel következnek be. Hasonlóan figyelniünk kell a geometriai valószínűség meghatározásakor is. Mivel jelen esetben a csúcsok egy köríven helyezkednek el, ezért szükséges, hogy ott bármely két azonos hosszúságú ívre „közel” azonos számú pont kerüljön. A koordináták előbbi megválasztásával azonban ezt az íveknél nem teljesítettük (ld. bal oldali ábra), mivel ott két azonos hosszúságú szakaszhoz két különböző hosszúságú ív tartozik. Akkor járunk el helyesen, ha az RND értékét adó $[0; 1[$ intervallumot a körívnek feleltetjük meg. Tehát egy véletlen csúcsot pl. az x tengelytől adott irányban fölmért $[0; 2\pi[$ szög jelöl ki a körvonalon.



A program feladata a következő: N számú kísérletet végez, és minden esetben megállapítja, hogy a kapott pontok egy hegyesszögű háromszög csúcsai-e. Megszámolja ezeket, és a kapott k darabszámot a végén elosztja N -nel.

Eljárás Hegy3koralon(N)

$k := 0$ (hegyesszögű háromszögek száma)

Ciklus $h := 1$ -től N -ig

alfa := $2\pi * \text{RND}(0, 1)$

ax := $\cos(\text{alfa})$: ay := $\sin(\text{alfa})$

beta := $2\pi * \text{RND}(0, 1)$

bx := $\cos(\text{beta})$: by := $\sin(\text{beta})$

gamma := $2\pi * \text{RND}(0, 1)$

cx := $\cos(\text{gamma})$: cy := $\sin(\text{gamma})$

Ha $\text{hegyes}(\text{ax}, \text{ax}, \text{bx}, \text{by}, \text{cx}, \text{cy})$ **akkor** $k := k + 1$

Ciklus vége

Ki: k/N

Eljárás vége

Az eljárásban szereplő $\text{hegyes}()$ függvény megállapítja, hogy a kapott koordinátákkal jellemzett háromszög hegyesszögű-e. Ehhez a szögek pontos nagyságát nem kell meghatározni, csak azt, hogy derékszögnél nagyobb, kisebb, vagy vele egyenlő-e. Például az A csúcsonál lévő szögről elég információt ad az $\overrightarrow{AB} \cdot \overrightarrow{AC}$ skaláris szorzat előjele. A függvény mindhárom skaláris szorzat pozitív értéke esetén ad igaz választ. Ennek elkészítését az olvasóra bízunk.

Változtassuk meg a feladatot úgy, hogy most a pontok egy körlapon található! Akkor járunk el helyesen, ha biztosítjuk, hogy a körlapon véletlenszerűen helyezkedjenek el a csúcsok. Minél több kísérletet végzünk, annál jobban teljesülnie kell, hogy a véletlenszerűen választott pontok „egyenletes sűrűséggel” vannak elszórva a körlapon. Pontosabban fogalmazva: legyen a körlap egy T_1 és T_2 területű részére eső véletlen pontok száma P_1 és P_2 . A kísérletek számának növelésével a P_1/P_2 hányados közelíti a T_1/T_2 értékét, miközben N értéke növekszik.

A körvonalnál láttuk, hogy a kapott $[0, 1[$ intervallumot kibővítettük $[0, 2\pi[$ intervallummá. Most területekről van szó, így jó gondolatnak tűnik, hogy egy véletlen csúcs x és y koordinátáját jelentse egy-egy RND érték. Ekkor azonban egy egység oldalú négyzeten oszlanak el egyenletesen a pontok. Ez nem jelent gondot, ha az egységsugarú körön kívüli pont esetén újra próbálkozunk, és egy másik véletlen számpárt kérünk. Ezzel persze tovább tart a program futása, de a megoldás a kitűzött feladatot oldja meg. A program ezek alapján a következő:

Eljárás Hegy3koralap(N)

$k := 0$ (hegyesszögű háromszögek száma)

Ciklus $h := 1$ -től N -ig

ax := $\text{RND}(0, 1)$: ay := $\text{RND}(0, 1)$

Ciklus amíg $\text{ax}^2 + \text{ay}^2 > 1$

ax := $\text{RND}(0, 1)$: ay := $\text{RND}(0, 1)$

Ciklus vége

... hasonlóan a másik két csúccsal

Ha hegyes(ax, ax, bx, by, cx, cy) akkor $k := k+1$

Ciklus vége

Ki: k/N

Eljárás vége

A feladat tovább bővíthető, pl. fölvehető, hogy a körvonalon vagy a körlapon véletlenszerűen választott pontok átlagosan mekkora kerületű, területű háromszöget határoznak meg. Ez utóbbi feladatok eredménye összetett számításokkal kapható, míg Monte-Carlo-módszerrel az eddigiekhez hasonlóan.

Lapunk korábbi számaiban foglalkoztunk a témával, a megjelent cikkek közül most Cserti László: *A munkára fogott véletlen*, I. részét¹ ajánlanánk, amelyben több példát láthatunk a módszer használatára.

Schmieder László



Informatikából kitűzött feladatok

I. 448 (É). Egy fanatikus kosárlabda-szurkoló előre szeretné megvenni a belépőjegyeket a bajnokság februártól májusig terjedő időszakának bizonyos mérkőzéseire. Kedvenc csapata a *Voros_Rokak*, de szívesen nézi a *Computerbonto* és a *Bohocok* meccseit is. Rendelkezésünkre áll, és honlapunkról letölthető a *naptar.txt* állomány, amely a mérkőzések adatait és a szurkolónak a belépők megvásárlására tervezett maximális pénzkeretét tartalmazza.

Az állomány első sorában a naptárban szereplő mérkőzések száma N ($1 \leq N \leq 100$) és a jegyvásárlásra szánt maximális pénzkeret P ($2000 \leq P \leq 50\,000$) található. Az ezt követő N sor az egyes mérkőzések négy adatát tartalmazza: a hazai csapat neve, az ellenfél csapatának neve, a nap sorszáma az éven belül és a jegyár. A mérkőzések egy napon belül sem kezdődnek egyszerre, a szervezők biztosítják, hogy elvileg mindegyik megtekinthető legyen.

Készítsünk programot *i448* néven, amely megoldja az alábbi feladatokat, amelyekkel a szurkoló jegyvásárlását segítjük. A képernyőre írást igénylő részfeladatok eredményének megjelenítése előtt írjuk a képernyőre a feladat sorszámát (például **4. feladat**:). A beolvasás előtt a várt tartalomra vonatkozó üzenetet jelenítsük meg.

1. Olvassuk be a *naptar.txt* állomány adatait és a következő feladatokat ezek alapján oldjuk meg.

¹<http://db.komal.hu/KomalHU/showpdf.phtml?tabla=Cikk&id=200459>.