Algoritmi probabilisti:
Algoritmi de tip Monte Carlo

- o complexitate de timp “buna”/”exacta”. Solutia oferita nu este neaparat cea mai buna, ci tine spre optim:
  Ex: Algoritm de tip MC pt aproximarea lui pi.

Algoritmi Las Vegas

- implica folosirea de selectie probabilistica (orice fel de element probabilistic) pentru a obține un rezultat exact intr-o complexitate de timp “probabil” buna.
  ex: QuickSort cu pivot ales aleator.
    - la final sigur obtinem vectorul sortat
    - complexitate: medie de O(nlogn) (practic si mai rapid) , worst case O(n^2)
    -

Ex1:
Fie un patrat de coordonate (0,0),(0,1),(1,1),(1,0)
fie un cerc C inscris in patrat:
    raza cercului: ½
    centrul cercului: (0.5,0.5)

daca generam (aleator si uniform) multe puncte in interiorul patratului, si vedem cate sunt in cerc, atunci raportul dintre punctele din cerc si numarul total de puncte va tinde la raportul dintre cerc si patrat, adica pi/4.

- o functie care sa va genereze aleator un punct in patrat
- o funcrie de distanta
- numarati punctele care sunt in cerc si facut raportul, iar apoi inmultiti cu 4.

Ex2:
    Acul lui Buffon:

    Avem un caiet dictando cu distanta dintre linii e 2*L si un ac de lungime L.
    Probabilitatea ca atunci cand dau drumul la ac, acesta sa intersecteze una dintre liniile caietului este de 2/pi.

Ex3:
Serii Taylor: - converg destul de repede la valoarea lui pi.

P(t) ->???-> P(t+1)

P(t): - criteriul elitist (cel mai bun individ, sau cei mai buni x% indivizi merg direct in P(t+1))

- pe locurile ramase libere (sa zicem ca raman *n* locuri libere), pe criteriul ruletei (sau turnelui) selectati *n* elemente din P(t) (aici pot fi incluse si duplicate, aici pot fi inclusi si indivizi care deja s-au calificat in P(t+1) fiind elitisti). Numim populatia asta intermediara P’(t)
- din P’(t) selectez un numar de indivizi astfel: parcurg fiecare individ, si cu o probabilitate de *pc* il marchez ca selectat pt procesul de crossing over.

- Fie X submultimea indivizilor selectati din P'(t). Aplic crossing over pe toate perechile din X (respectiv daca |X| impar, iau ultimii 3 cromozomi "la pachet"). Se obtine o noua multime X', aceasta va inlocui pe X in P'(t), astfel se obtine P''(t).
- peste P''(t) aplica operatorul de mutatie: - ori mutatie rara (fiecare cromozom are o probabilitate mica de a fi selectat, celor marcati ca fiind selectat lise va schimba o gena de pe o pozitie aleatoare)
    - mutatie "regular": itereze prin toti indivizii. pt fiecare individ iterez prin gene si exista o mica probabilitate ca la o gea prin care iterez sa ii si schimb valoarea.

Asa obtin P(t+1)

0.10 0.15 0.10 0.30 0.15 0.2
[0,0.10),[0.10,0.25);[0.25,0.35);[0.35,0.65);[0.65,0.80);[0.8,1)
0.457