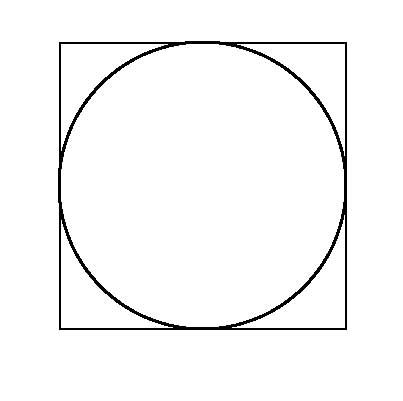
Algoritmi Probabilisti:

Algoritmi Las Vegas:

-vom obtine mereu solutia cautata! Factorul de probabilitate va influenta (in mod pozitiv) numarul de pasi efectuati de catre algoritm. Ex: QuickSort cu pivot ales aleator.

Faptul ca pivotul este ales aleator - implca o complexitate medie de O(n log n); worst case O(n2).

Algoritmi Monte Carlo: - genereaza o solutie aproape de cea dorita, intr-un mod determinist si timp “rapid” (de obicei liniar).



Ex: un algoritm foarte simplu pentru a aproxima valoarea lui pi.

Avem un patrat cu colturile (0,0) (0,1) (1,1) (1,0)

avem un cerc inscris in acest patrat:

centrul: (½,½ )

raza: ½

Generez aleator uniform distribuit *n* puncte in interiorul patratului.

Unele puncte se vor fi nimerit si in cerc. Fie p - numarul de puncte care sunt si in cerc.

Pentru un *n* foarte mare, la cat va tinde raportul p/n?

Va tinde la raportul suprafetelor (cerc vs patrat)

p/n ~> pi/4

Structura algoritmului:

* o structura pentru a retine puncte in plan (pereche de floats)
* o functie care sa genereze aleator si uniform puncte in patrat
* o functie care calculeaza distanta dintre 2 puncte. Pentru a verifica daca un punct generat este sau nu in cerc.
* repet de un numar determinat de ori (primit ca input) - generez punct in patrat, vad daca acest punct este si in cerc, incrementez un contor.
* la final fac raportul dintre contor si n iar apoi inmultesc rezultatul cu 4, obtinand o aproximare pentru pi.