Université Larbi Ben Mhidi — Niveau : 1ère année Master AD

Département MI — Module : Méthodes algorithmiques

Année : 2024-2025

Solution de TD 4 : Les méthodes probabiliste

Exercice 01 :

```
import random

def calculer_surface_approximation(nb_points, rayon):
    points_dans_cercle = 0
    for _ in range(nb_points):
        coord_x = random.uniform(-rayon, rayon)
        coord_y = random.uniform(-rayon, rayon)


        if coord_x**2 + coord_y**2 <= rayon**2:
            points_dans_cercle += 1
    proportion = points_dans_cercle / nb_points
    surface_cercle = proportion * (4 * rayon**2)
    return surface_cercle
nombre_points = 100000
rayon_cercle = 1
surface_estimee = calculer_surface_approximation(nombre_points, rayon_cercle)
print(f"Estimation de l'aire du cercle : {surface_estimee}")
```

EXO 2

```
import random

def est_premier_las_vegas(n, essais_max=100):
    if n < 2:
        return False
    if n in (2, 3):
        return True
    if n % 2 == 0:
        return False

    for _ in range(essais_max):
        a = random.randint(2, n - 2)
        if pow(a, n - 1, n) != 1
            return False
    return True


    print(f"{nombre} est composé.")
```

EXO 3

```
def recherche_probabiliste(A, x):
    indices = random.sample(range(len(A)), int(len(A) ** 0.5))
    for i in indices:
        if A[i] == x:
            return True
    return False
```

EXO 4

```
import random

def max_probabiliste(A):
    n = len(A)
    m = n // 2

    sous_ensemble = random.sample(A, m)

    maximum = max(sous_ensemble)

    return maximum
```