Mathématiques & Informatique - F. Gaunard
http://frederic.gaunard.com
Lycée Voltaire, Paris 11e.

Préparation à l'oral

Math-Info : Math 2
Semaine du 8 Juin

Sujet OB-MATH2-7

Exercice de mathématiques

Soient n un entier naturel impair et P un polynôme de $\mathbb{R}[X]$ ayant la propriété $(\mathscr{P})$ suivante :

$$(X^n + 1)\,P(X) = P(X^2).$$

1. Dire si le polynôme $X^n - 1$ a la propriété $(\mathscr{P})$.
2. Déterminer le degré du polynôme P.
3. Soit ω une racine n-ième de -1. Montrer que $-\omega$ est racine de P.
4. Quels sont les polynômes ayant la propriété $(\mathscr{P})$?

Exercice d'informatique

On considère un jeu de 32 carte formé de couples de 8 valeurs de 4 «couleurs» données par `valeurs= ["7","8","9","10","V","D","R", "A"]` et `couleurs= ["T", "K", "C", "P"]`.

On distribue au hasard une «main», c'est-à-dire 5 cartes distinctes. On s' intéresse à des mains particulières :

- ✗ les «*couleurs*» (5 cartes de même « couleur») ;
- ✗ les «*quintes*» (5 cartes de valeurs qui se suivent);
- ✗ les «*quintes floches*» (5 cartes de même «couleur» et de valeurs qui se suivent).

1. Construire l'ensemble cartes des 32 cartes à partir des deux listes `valeurs` et `couleurs`, chaque carte étant représentée par la paire `[valeur, couleur]`. Vérifier que le nombre de cartes obtenu est correct.

2. Écrire une fonction `tirer_main` sans argument qui renvoie une liste de 5 cartes distinctes tirées au hasard. On pourra utiliser la fonction `sample` du module `random`.

3. Écrire une fonction `test_couleur` d'argument une main et qui renvoie un booléen indiquant si cette main est une «*couleur*» ou pas.

4. Déterminer la liste `quintes` de toutes les suites possibles de 5 valeurs d'une *quinte*. En déduire une fonction `test_quinte` d'argument une main et qui renvoie un booléen indiquant si cette main est une «*quinte*» ou pas.
Indication : pour comparer deux listes de valeurs indépendamment de leur ordre, on pourra utiliser la fonction intrinsèque `sorted`.

5. À partir de 50000 tirages aléatoires de mains, estimer la probabilité d'obtenir une «*couleur*», celle de tirer une «*quinte*» et celle de gagner une «*quinte floche*».

6. Comparer ces estimations avec les probabilités obtenues par dénombrement.
On pourra utiliser la fonction `comb` du module `scipy.special`, ou faire autrement.

Solution du sujet OB-MATH2-7

Math (Math II 2025)

1. On a d'une part $(X^n+1)(X^n-1)=X^{2n}-1$ et $P(X^2)=(X^2)^n-1=X^{2n}-1$, d'autre part. Donc, X^n-1 vérifie la propriété $(\mathscr{P})$.

2. Le polynôme nul vérifie clairement la propriété $\mathscr{P}$. Soit donc P un polynôme non nul vérifiant la propriété $\mathscr{P}$ et soit d son degré. On a :
$$\deg((X^n+1)P(X))=\deg(P(X^2))$$
Donc,
$$n+d=2d$$
ce qui donne
$$d=n$$
Le degré du polynôme P est donc n.

3. Soit ω une racine n-ième de -1, c'est-à-dire $\omega^n=-1$. En utilisant la propriété $\mathscr{P}$, on a :
$$(X^n+1)P(X)=P(X^2)$$
En évaluant l'identité précédente en ω, on obtient :
$$(\omega^n+1)P(\omega)=P(\omega^2)$$
Comme $\omega^n=-1$, on a $P(\omega^2)=0$. En évaluant maintenant l'identité précédente en $-\omega$, on obtient :
$$((-\omega)^n+1)\,P(-\omega)=P(\omega^2)$$
Compte tenu de l'imparité de n, $(-\omega)^n=(-1)^n\omega^n=-(-1)=1$, ce qui, avec la relation précédente, donne:
$$2P(-\omega)=P(\omega^2)=0$$
Ainsi, $-\omega$ est racine de P.

4. On vient de voir que les opposées des n racines n-ième de -1 sont racines de P. Comme celui-ci est de degré n, il n'en admet pas d'autres. En remarquant que, comme n est impair, ω est racine n-ième de -1 si et seulement si $-\omega$ est racine n-ième de 1, on déduit que le racines de P sont exactement les racines n-ième de 1; par conséquent, les polynômes vérifiant la propriété $(\mathscr{P})$ sont de la forme :
$$P(X)=\mu\prod_{k=0}^{n-1}\left(X-\exp\left(\frac{2ik\pi}{n}\right)\right)=\mu\,(X^n-1)$$
où μ est une constante réelle arbitraire.

   Bien entendu tous les polynômes de la forme précédente conviennent comme on l'a vu à la question 1.

Info (Math II 2021)

1.
```python
valeurs = ["7","8","9","10","V","D","R","A"]
couleurs = ["T","K","C","P"]

cartes = [[v, c] for v in valeurs for c in couleurs]

print(len(cartes))   # 32
```

2.
```python
from random import sample

def tirer_main():
    return sample(cartes, 5)
```

3.
```python
def test_couleur(main):
    c = main[0][1]
    for carte in main:
        if carte[1] != c:
            return False
```

```python
    return True
```

4. ```python
quintes = [valeurs[i:i+5] for i in range(len(valeurs)-4)]

def test_quinte(main):
    vals = [carte[0] for carte in main]
    return sorted(vals, key=valeurs.index) in quintes
```

5. La fréquence d'apparition d'un évènement lors d'un grand nombre de répétitions de l'expérience fournit une valeur approchée de la probabilité théorique de l'évènement (cela découle de la loi faible des grands nombres).

```python
N = 50000

nb_couleur = 0
nb_quinte = 0
nb_quinte_floche = 0

for _ in range(N):
    main = tirer_main()

    couleur = test_couleur(main)
    quinte = test_quinte(main)

    if couleur:
        nb_couleur += 1
    if quinte:
        nb_quinte += 1
    if couleur and quinte:
        nb_quinte_floche += 1

print("P(couleur) ?", nb_couleur / N)
print("P(quinte) ?", nb_quinte / N)
print("P(quinte floche) ?", nb_quinte_floche / N)
```

6. ```python
from math import comb

total = comb(32, 5)

nb_couleurs = 4 * comb(8, 5)
p_couleur = nb_couleurs / total

nb_quintes = 4 * (4**5)
p_quinte = nb_quintes / total

nb_quintes_floches = 4 * 4
p_quinte_floche = nb_quintes_floches / total

print("P(couleur) =", p_couleur)
print("P(quinte) =", p_quinte)
print("P(quinte floche) =", p_quinte_floche)
```