qui donne

$$\begin{cases} x = y = \dfrac{1}{xy} \end{cases}$$

ainsi $x = y = 1$. Le point $(1,1)$ est l'unique point critique de f sur $]0,+\infty[^2$.

— Nature du point critique :

L'inégalité de la question précédente appliquée pour $a = x$, $b = y$ et $c = \frac{1}{xy}$, donne

$$\sqrt[3]{abc} = 1 \leq \frac{a+b+c}{3} = \frac{1}{3} f(x,y)$$

On en déduit : $\forall (x,y) \in \,]0,+\infty[^2 \; f(x,y) \geq f(1,1) = 3$ donc $(1,1)$ est un minimum global .

PROBLEME

Un peu d'arithmétique avec la fonction zêta de Riemann

Partie I - lgorithmique : calcul de zêta aux entiers pairs

Q5.

```
def f ctorielle(n):
    P = 1
    for i in r nge(2,n+1):
            P*=i
    return P
```

Q6.

```
def binom(n,p):
    if not (0 <= p <= n) : return 0
    return f ctorielle(n)//(f ctorielle(p)*f ctorielle(n-p))
```

Remarque : la version proposée `binom(n,p)` est la version naïve qui effectue $2n$ opérations de multiplications. Le nombre exacte de multiplications dépend de la plage des valeurs de `i` utilisée dans la boucle `for`. Dans notre cas de `r nge(2,n+1)` la boucle effectue `n-1` multiplications pour tout $n \geq 2$. Donc :

`f ctorielle(n)` effectue n-1 multiplications

`f ctorielle(p)` effectue p-1 multiplications

`f ctorielle(n-p)` effectue n-p-1 multiplications

et il faut y ajouter une multiplication `f ctorielle(p)*f ctorielle(n-p)`

D'où le nombre total de multiplications = (n-1) + (p-1) + (n-p-1) + 1 = 2n-2.

Donc `binom(30,10)` va effectuer 58 multiplications.

On peut réduire le nombre de multiplications en utilisant l'expression

`(n*(n-1)*(n-2)*....*(n-(p-1)))//f ctorielle(p)`

pour calculer le binomial, qui effectue uniquement 2p multiplications. Et par conséquent `binom(30,10)` va effectuer 20 multiplications au maximum (19 selon notre code).

Le type du résultat de "`return f ctorielle(n)/f ctorielle(p)*f ctorielle(n-p)`" est float.

Q7.

```
def binom_rec(n,p):
        if p == 0 : return 1
        return int(n/p*binom_rec(n-1,p-1))
```

Remarque : il ne faut pas utiliser l'expression `return n//p * binom_rec(n-1,p-1)`

Q8.

La question demande d'écrire une version non récursive de `bernoulli(n)`.

En effet, il est plus facile d'écrire une version récursive de `bernoulli(n)` qui reflète exactement l'expression récurrente

donnée dans l'énoncé. Mais elle aura une complexité exponentielle qui est non pratique et qui nécessite un temps imaginaire pour

s'exécuter pour des valeurs de $n \geq 30$.

Cette version qui est raisonnablement inacceptée est la suivante :

```
def bernoulli(n):
        if n == 0:return 1
        S = 0
        for k in r nge(n):
                S+ = binomi l(n+1,k)*bernoulli(k)
        return-S/(n+1)
```

La version non récursive qui a une complexité raisonnable (polynomiale) consiste à stocker les nombres de Bernoulli b_i dans une liste `b` de taille $n+1$, telle que pour tout $0 \leq i \leq n$ `b[i]`=le nombre de Bernoulli b_i. insi, On peut utiliser les termes de la suite des nombres de Bernoulli $(b_k)_{k<i}$ stockés dans la liste `b` pour calculer le terme b_i.

Il suffit enfin de renvoyer la dernière valeur de la liste b qui contient la valeur de b_n.

```
def bernoulli(n):
        b=[1] #b =[b0]
        for i in r nge(1,n+1):
                S = 0
                for k in r nge(i):
                        S+ = binomi l(i + 1,k)*b[k] #b[k]=b
                b. ppend(-S/(i + 1))
        return b[n]
```

Il serait plus pratique de renvoyer la liste `b` qui va contenir tous les nombres de Bernoulli `b` tel que $0 \leq k \leq n$, ce qui nous

permettrait par la suite de récupérer chaque `b` en O(1).

Il est à noter qu'on a le droit d'utiliser la fonction `binomi l(n,p)` sans la définir. Mais nous avons jugé important de l'ajouter

dans cette proposition de correction.

La fonction `binomi l(n,p)` plus optimale qu'on va proposer se base sur l'expression
(n*(n-1)*(n-2)*....*(n-(p-1))) //factorielle(p)

```
def binomi l(n,p):
        # on c lculer  d' bord prod =n*(n-1)*(n-2)*....*(n-(p-1))
        prod = 1
        for k in r nge(p):
                prod* = (n-k)
        return prod//f ctorielle(p)
```

Partie II - Généralités sur la fonction zêta

Pour tout $n \in \mathbb{N}^*$, on note f_n la fonction définie sur $]1, +\infty[$ par :

$$f_n(x) = \frac{1}{n^x}$$

Q9. Soit $a > 1$, prenons $\quad \in]1, a[$ de sorte que $\dfrac{\ln n}{n^a} \underset{n \to +\infty}{=} o(\dfrac{1}{n})$, comme$\sum \dfrac{1}{n}$ converge alors $\sum \dfrac{\ln n}{n^a}$ est convergente.

Q10. On a$\sum f_n$ converge simplement sur $]1, +\infty[$. f_n est de classe $\mathcal{C}^1$sur $]1, +\infty[$ et $f_n'(x) = \dfrac{\ln n}{n^x}$.
Soit $a > 1$, on a $\sup\limits_{x \in \ a, +\infty} |f_n'(x)| = \dfrac{\ln n}{n^a}$, or la série $\sum \frac{\ln n}{n}$ est convergente alors la série de fonctions $\sum f_n'$ converge normalement , et uniformément sur $[a, +\infty[$. D'après le théorème de dérivabilité des séries de fonctions, la fonction ζ est de classe $\mathcal{C}^1$ sur $[a, +\infty[$,ceci étant vrai pour tout $a > 1$ donc ζ est de classe $\mathcal{C}^1$ sur $]1, +\infty[$ et

$$\forall x \in]1, +\infty[\,, \ \zeta'(x) = \sum_{n=1}^{+\infty} \frac{\ln n}{n^a}$$

Les termes de la série, définissant ζ' , sont négatifs donc ζ' est négatif et ζ est décroissante sur $]1, +\infty[$.

Q11. On a $\lim\limits_{x \to 1^+} f_n(x) = \frac{1}{n}$. Si $\sum f_n$ converge uniformément sur $]1, +\infty[$, le théorème d'interversion de lim et $\sum$ donne $\sum \frac{1}{n}$converge , ce qui est absurde , donc $\sum f_n$ ne converge pas uniformément sur $]1, +\infty[$.
on a convergence normale et uniforme de $\sum f_n$ sur $[a, +\infty[$ pour tout $a > 1$.

Q12. $\sum f_n$ converge uniformément sur $[2, +\infty[$ et $\lim\limits_{x \to +\infty} f_n(x) = \begin{cases} 0 \text{ si } n \geq 2 \\ 1 \text{ si } n = 1 \end{cases}$, le théorème d'interversion de lim et $\sum$ donne $\lim\limits_{x \to +\infty} \zeta(x) = 1$.

Q13. Soit $x > 1$ et $k \in \mathbb{N}^*$, pour $t \in [k, k+1]$, on a $\frac{1}{(k+1)^x} \leq \frac{1}{t^x} \leq \frac{1}{k^x}$. On intègre cette relation , entre k et $k+1$ on obtient ;

$$\frac{1}{(k+1)^x} \leq \int_k^{k+1} \frac{dt}{t^x} \leq \frac{1}{k^x}$$

on fait la somme pour tout $k \geq 1$, on obtient

$$\zeta(x) \quad 1 \leq \int_1^{+\infty} \frac{dt}{t^x} \leq \zeta(x) \quad .$$

insi

$$I(x) \leq \zeta(x) \leq I(x) + 1$$