



NOMBRE:

NÚM.:

PRÁCTICAS: Backtracking

10 puntos

Haz una función python que reciba un número natural N y, mediante backtracking, imprima todas las permutaciones de los números del 0 al $N - 1$ de modo que el número i no ocupe el índice i . Por ejemplo, para $N = 3$ mostraría:

[1, 2, 0]
[2, 0, 1]

Para $N = 4$ mostraría:

[1, 0, 3, 2]
[1, 2, 3, 0]
[1, 3, 0, 2]
[2, 0, 3, 1]
[2, 3, 0, 1]
[2, 3, 1, 0]
[3, 0, 1, 2]
[3, 2, 0, 1]
[3, 2, 1, 0]

Nota: no hay que generar todas las permutaciones y luego elegir las que cumplen la condición, hay que evitar generar soluciones parciales no prometedoras.

Solución:

```
def permutaciones(N):  
    sol = []  
    def backtracking(indice):  
        if indice == N:  
            yield sol.copy()  
        else:  
            for valor in range(N):  
                if valor != indice and valor not in sol:  
                    sol.append(valor)  
                    yield from backtracking(indice+1)  
                    sol.pop()  
  
    yield from backtracking(0)  
  
if __name__ == "__main__":  
    for x in permutaciones(3):  
        print(x)
```