

rademacher_cos.py

```
1 """
2 Fredrik Johansson
3 https://gist.github.com/fredrik-johansson/7c2711887811ef9f2d7038b8451a4e63
4 MIT license, 2020
5
6 Reduced to the real 'cos' case and type annotations
7 added by Peter Luschny (2024).
8 """
9
10 from math import sinh, cosh, sqrt, cos, pi
11 from functools import cache
12 from flint import fmpz, fmpq
13
14 @cache
15 def dedekind_sum(
16     r: int,
17     k: int
18 ) → None | float:
19
20     if fmpz(r).gcd(k) ≠ 1:
21         v = None
22     else:
23         u = fmpq.dedekind_sum(r, k)
24         p = int(u.p)
25         q = int(u.q)
26         v = float(p) / float(q)
27     return v
28
29
30 def rademacher_cos(n: float, k: int) → float:
31     s = 0.0
32     for r in range(k):
33         v = dedekind_sum(r, k)
34         if v is not None:
35             s += cos(pi * (v - 2.0 * n * r / k))
36     return s
37
38
39 def bessel32(x: float) → float:
40     s = sinh(x)
41     c = cosh(x)
42     return (2 * c - 2 * s / x) / (sqrt(2 * pi * x))
43
44
45 def rademacher(
46     n: float,
47     tol: float = 1e-5,
48     consecutive: int = 20
49 ) → float:
50
51     n = float(n)
52     if n == 0.0:
53         return float(1.0)
54     k = 1
55     s = 0.0
56     b = 10**9
57     c = sqrt(2 * (n - 1.0 / 24) / 3)
```

```

58 while k < b:
59     a = rademacher_cos(n, k)
60     x = (pi / k) * c
61     t = a * bessell32(x) / k
62     if abs(t) < tol:
63         b = min(b, k + consecutive)
64     else:
65         b = 10**9
66     s += t
67     k += 1
68 return 2.0 * s * pi / float(24 * n - 1) ** 0.75
69
70
71 def part_int(n: float) → int:
72     return round(rademacher(n))
73
74
75 if __name__ == '__main__':
76
77     def demo() → None:
78         print("partition function at n:")
79         print([part_int(n) for n in range(20)])
80         print("partition function at n + 1/2:")
81         print([part_int(n + 1/2) for n in range(20)])
82         print("partition function at n and n + 1/2:")
83         print([part_int(p//2 + (p%2)/2) for p in range(25)])
84
85     demo()

```