As warm up, we will repeat some of the assignments from the previous quarter, but this time you *must* implement everything in python. Answering these questions does not require extensive familiarity with the large number of packages in the python runtime: we will only use the package `random`. But feel free to look around.

1. Recall the pseudocode for MERGE-SORT:

Algorithm 1: MERGE-SORT(S, p, r)

```
1 if p < r then
2 |   q ← ⌊(p + r)/2⌋
3 |   MERGE-SORT(S, p, q)
4 |   MERGE-SORT(S, q + 1, r)
5 |   MERGE(S, p, q, r)
```

   (a) Implement MERGE-SORT and MERGE, its workhorse, in python.

   (b) Write a driver that invokes it with $S = (5, 2, 4, 6, 1, 3)$.

   (c) Build a container with 10^6 random numbers. Sort it using your implementation.

2. Recall the definition for the dilogarithm in terms of the integral

$$\mathrm{Li}_2(z) := -\int_0^z dz' \, \frac{\log(1 - z')}{z'} \tag{1}$$

   (a) Implement a function `dilog` that accepts a real number z and an integer N, and returns an approximation to Eq. 1 using the midpoint rule.

   (b) Build a table of the value, the error and the amount of time it takes to compute $\mathrm{Li}_2(1)$ and $\mathrm{Li}_2(-1)$, for $N \in \{10, 10^3, 10^6, 10^9\}$.

3. Use Monte Carlo integration to compute an approximation to the value of π by computing the area of the upper right quadrant of a unit circle centered at the origin, as shown below.