Math1402: Python project - Ants

# Team

Jasper Alizond (??)

Julia Ronneberger (16078856)

# Output of Program (using N = 10000)

(3,4,5): P(a) = 0.2804 P(b) = 0.3277 P(c) = 0.3919

(5,12,13): P(a) = 0.2127 P(b) = 0.3814 P(c) = 0.4059

(8,15,17): P(a) = 0.2378 P(b) = 0.3565 P(c) = 0.4057

(7,24,25): P(a) = 0.1739 P(b) = 0.4083 P(c) = 0.4178

(20,21,29): P(a) = 0.3005 P(b) = 0.3046 P(c) = 0.3949

(12,35,37): P(a) = 0.1912 P(b) = 0.3917 P(c) = 0.4171

(9,40,41): P(a) = 0.1507 P(b) = 0.4189 P(c) = 0.4304

(28,45,53): P(a) = 0.2672 P(b) = 0.3484 P(c) = 0.3844

(11,60,61): P(a) = 0.1348 P(b) = 0.4344 P(c) = 0.4308

(16,63,65): P(a) = 0.1686 P(b) = 0.4052 P(c) = 0.4262

(33,56,65): P(a) = 0.2495 P(b) = 0.364 P(c) = 0.3865

(48,55,73): P(a) = 0.294 P(b) = 0.3208 P(c) = 0.3852

(13,84,85): P(a) = 0.1241 P(b) = 0.4365 P(c) = 0.4394

(36,77,85): P(a) = 0.2305 P(b) = 0.3661 P(c) = 0.4034

(39,80,89): P(a) = 0.219 P(b) = 0.3759 P(c) = 0.4051

(65,72,97): P(a) = 0.2916 P(b) = 0.3136 P(c) = 0.3948

(20,99,101): P(a) = 0.1488 P(b) = 0.4179 P(c) = 0.4333

(60,91,109): P(a) = 0.2596 P(b) = 0.3501 P(c) = 0.3903

(15,112,113): P(a) = 0.1084 P(b) = 0.4479 P(c) = 0.4437

(44,117,125): P(a) = 0.2002 P(b) = 0.3903 P(c) = 0.4095

(88,105,137): P(a) = 0.2801 P(b) = 0.325 P(c) = 0.3949

(17,144,145): P(a) = 0.1037 P(b) = 0.445 P(c) = 0.4513

(24,143,145): P(a) = 0.1204 P(b) = 0.4412 P(c) = 0.4384

(51,140,149): P(a) = 0.1923 P(b) = 0.3934 P(c) = 0.4143

(85,132,157): P(a) = 0.2612 P(b) = 0.3446 P(c) = 0.3942

(119,120,169): P(a) = 0.3088 P(b) = 0.3111 P(c) = 0.3801

(52,165,173): P(a) = 0.1862 P(b) = 0.3974 P(c) = 0.4164

(19,180,181): P(a) = 0.0962 P(b) = 0.4471 P(c) = 0.4567

(57,176,185): P(a) = 0.188 P(b) = 0.3954 P(c) = 0.4166

(104,153,185): P(a) = 0.2678 P(b) = 0.3429 P(c) = 0.3893

(95,168,193): P(a) = 0.2487 P(b) = 0.3457 P(c) = 0.4056

(28,195,197): P(a) = 0.1146 P(b) = 0.4408 P(c) = 0.4446

(84,187,205): P(a) = 0.2177 P(b) = 0.3734 P(c) = 0.4089

(133,156,205): P(a) = 0.2983 P(b) = 0.3122 P(c) = 0.3895

(21,220,221): P(a) = 0.0887 P(b) = 0.4589 P(c) = 0.4524

(140,171,221): P(a) = 0.2799 P(b) = 0.3206 P(c) = 0.3995

(60,221,229): P(a) = 0.1716 P(b) = 0.4044 P(c) = 0.424

(105,208,233): P(a) = 0.2319 P(b) = 0.3647 P(c) = 0.4034

(120,209,241): P(a) = 0.2339 P(b) = 0.3635 P(c) = 0.4026

(32,255,257): P(a) = 0.1109 P(b) = 0.4426 P(c) = 0.4465

(23,264,265): P(a) = 0.08 P(b) = 0.4612 P(c) = 0.4588

(96,247,265): P(a) = 0.2061 P(b) = 0.3923 P(c) = 0.4016

(69,260,269): P(a) = 0.1682 P(b) = 0.408 P(c) = 0.4238

(115,252,277): P(a) = 0.2234 P(b) = 0.3717 P(c) = 0.4049

(160,231,281): P(a) = 0.2673 P(b) = 0.3398 P(c) = 0.3929

(161,240,289): P(a) = 0.265 P(b) = 0.3415 P(c) = 0.3935

(68,285,293): P(a) = 0.1573 P(b) = 0.4133 P(c) = 0.4294

Triple with smallest probability that ant leaves at hypotenuse:

(119, 120, 169): 0.3801

Triple with largest probability that ant leaves at hypotenuse:

(23, 264, 265): 0.4588

# Histogram

# Output for (3,4,5) with different N

Exact probability is 0.3916721504

|  |  |  |
| --- | --- | --- |
| **N** | **Probability that ant leaves at hypotenuse** | **Error (difference between my prob and exact prob)** |
| 1,000 | 0.382 | 0.0096721504 |
| 10,000 | 0.3919 | 0.0002278496 |
| 100,000 | 0.38993 | 0.0017421504 |
| 1,000,000 | 0.391972 | 0.0002998496 |
| 10,000,000 | 0.3917863 | 0.0001141496 |
| 100,000,000 | 0.39162506 | 0.0000470904 |

To get a result that is precise up to 10 digits after the decimal point, we would need N to be **at least** 10,000,000,000.

With N = 100,000,000, the algorithm already took about 10 minutes, so for N = 10,000,000,000 we should consider a different algorithm.