Вывод: из графиков видно, что при условии, что m значительно меньше n лучшим из трёх алгоритмов является алгоритм бинарного поиска в каждой строке, а худшим диагональный поиск. При 5 < m < 8 лучшим становится диагональный алгоритм с экспоненциальным ускорением. Но при m >= 8, алгоритм бинарного поиска в каждой строке становится худшим из трёх по времени выполнения, лучшим остаётся диагональный алгоритм с экспоненциальным ускорением. Также следует заметить, что на первом типе генерации данных(a[i][j] = ((n/m) \* i + j) \* 2, target = 2 \* n – 1) все алгоритмы работают медленнее, чем на втором типе генерации данных(a[i][j] = ((n/m) \* i \* j) \* 2, target = 16 \* n – 1). Диагональный алгоритм с экспоненциальным ускорением лучше работает на 2 типе данных, так как он более эффективен при поиске в “длину”(благодаря экспоненциальному поиску).