UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

RESEARCH TRACK 2

First Assignment

Statistical Analysis on the First Assignment (RT1)

Author:

Galvagni Gianluca

Student ID:

s5521188

Professor:

Carmine Tommaso Recchiuto

May 22, 2023

Contents

1 Assignment description

Statistical Analysis of Token Placement Algorithms.

1.1 Objective

The objective of this assignment is to perform a statistical analysis to compare two different implementations of token placement algorithms. Specifically, you will analyze the performance of your own implementation and that of one of your colleagues when silver and golden tokens are randomly placed in the environment. The analysis will aim to determine which algorithm performs better based on the given performance metrics. Additionally, you will carefully plan the experiments, choose a suitable statistical approach, and write a report summarizing the findings.

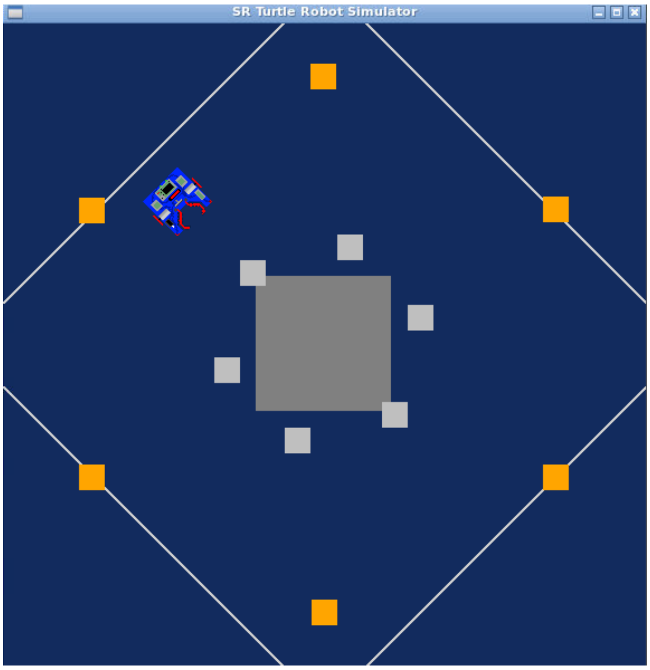

Figure 1:Starting point first RT1 assignment.

1.2 Task Description

1. **Hypotheses Formulation:**

   ◇ ***Null Hypothesis (H0)*** represents the default assumption or the statement of no effect or no difference. It assumes that there is no significant difference or relationship between the variables under investigation.

   ◇ ***Alternative Hypothesis (Ha)*** opposes the null hypothesis and suggests that there is a significant difference or relationship between the variables. It represents the possibility of an effect or difference existing.

2. **Experimental Setup:**

   → ***Experiment Types*** Conduct experiments with silver and golden tokens placed in two different scenarios:

   - Tokens placed on the two circles: Test how well the algorithms perform when tokens are positioned on the two circles.
   - Tokens randomly placed in the environment: Assess the algorithms' performance when tokens are randomly distributed throughout the environment.

   → ***Number of Repetitions*** Perform each experiment type 50 times to obtain sufficient data for statistical analysis.

   → ***Number of tokens*** For each experiment i will use 3 gold tokens and 3 silver tokens.

3. **Performance Evaluation Metrics:** As performance evaluators, consider the following metrics:

   – **Average Time Required** Measure the average time taken by each implementation to complete the task of retrieving all tokens. This metric will provide insights into the efficiency of the algorithms in terms of time complexity.

   – **Success/Failure Count** Record the number of successful and failed attempts to retrieve tokens using each algorithm. A successful attempt is when all tokens are retrieved, while a failed attempt is when one or more tokens remain uncollected. This metric will help evaluate the effectiveness and robustness of the algorithms.

4. **Statistical Analysis:**

   a. ***Hypotheses Made*** Clearly state the null and alternative hypotheses formulated for the statistical analysis.

   b. ***Description and Motivation of Experimental Setup*** Provide a detailed description of the experimental setup, including the two different scenarios for token placement and the number of repetitions for each experiment type. Explain the rationale behind these choices.

   c. ***Results and Discussion of Them with Statistical Analysis*** Present the collected data, including the average time required and the success/failure counts for each algorithm in both scenarios. Analyze the results obtained from the statistical tests, discussing any significant differences observed between the algorithms.

   d. ***Conclusion*** Summarize the overall findings and state whether the hypothesis was proven or rejected based on the statistical analysis.

2 Hypotheses Made

In order to evaluate the performance of my token placement algorithm compared to my colleague's algorithm, I have formulated the following hypotheses:

2.1 Null Hypothesis (H0)

There is no significant difference in performance between my token placement algorithm and my colleague's algorithm.

I propose the null hypothesis, assuming that both token placement algorithms yield similar performance results. This hypothesis suggests that any observed differences in performance between the two algorithms are merely due to random chance or other factors unrelated to the algorithms themselves. Essentially, I am stating that my algorithm and my colleague's algorithm perform equally well.

2.2 Alternative Hypothesis (Ha)

There is a significant difference in performance between my token placement algorithm and my colleague's algorithm.

On the other hand, I propose the alternative hypothesis, which challenges the null hypothesis. I believe that there exists a significant difference in performance between the two token placement algorithms. This hypothesis suggests that the observed differences in performance cannot be explained by random chance alone but are instead attributable to the inherent characteristics or design of the algorithms. In other words, either my algorithm or my colleague's algorithm outperforms the other.

By formulating these hypotheses, I aim to conduct a statistical analysis to determine whether the observed performance difference, if any, is statistically significant. The goal is to collect empirical evidence that either supports the null hypothesis or leads me to reject it in favor of the alternative hypothesis. This analysis will provide valuable insights into the relative effectiveness of my token placement algorithm compared to my colleague's algorithm.

***t* Table**

cum. prob	$t_{.50}$	$t_{.75}$	$t_{.80}$	$t_{.85}$	$t_{.90}$	$t_{.95}$	$t_{.975}$	$t_{.99}$	$t_{.995}$	$t_{.999}$	$t_{.9995}$
one-tail	**0.50**	**0.25**	**0.20**	**0.15**	**0.10**	**0.05**	**0.025**	**0.01**	**0.005**	**0.001**	**0.0005**
two-tails	**1.00**	**0.50**	**0.40**	**0.30**	**0.20**	**0.10**	**0.05**	**0.02**	**0.01**	**0.002**	**0.001**
df											
1	0.000	1.000	1.376	1.963	3.078	6.314	12.71	31.82	63.66	318.31	636.62
2	0.000	0.816	1.061	1.386	1.886	2.920	4.303	6.965	9.925	22.327	31.599
3	0.000	0.765	0.978	1.250	1.638	2.353	3.182	4.541	5.841	10.215	12.924
4	0.000	0.741	0.941	1.190	1.533	2.132	2.776	3.747	4.604	7.173	8.610
5	0.000	0.727	0.920	1.156	1.476	2.015	2.571	3.365	4.032	5.893	6.869
6	0.000	0.718	0.906	1.134	1.440	1.943	2.447	3.143	3.707	5.208	5.959
7	0.000	0.711	0.896	1.119	1.415	1.895	2.365	2.998	3.499	4.785	5.408
8	0.000	0.706	0.889	1.108	1.397	1.860	2.306	2.896	3.355	4.501	5.041
9	0.000	0.703	0.883	1.100	1.383	1.833	2.262	2.821	3.250	4.297	4.781
10	0.000	0.700	0.879	1.093	1.372	1.812	2.228	2.764	3.169	4.144	4.587
11	0.000	0.697	0.876	1.088	1.363	1.796	2.201	2.718	3.106	4.025	4.437
12	0.000	0.695	0.873	1.083	1.356	1.782	2.179	2.681	3.055	3.930	4.318
13	0.000	0.694	0.870	1.079	1.350	1.771	2.160	2.650	3.012	3.852	4.221
14	0.000	0.692	0.868	1.076	1.345	1.761	2.145	2.624	2.977	3.787	4.140
15	0.000	0.691	0.866	1.074	1.341	1.753	2.131	2.602	2.947	3.733	4.073
16	0.000	0.690	0.865	1.071	1.337	1.746	2.120	2.583	2.921	3.686	4.015
17	0.000	0.689	0.863	1.069	1.333	1.740	2.110	2.567	2.898	3.646	3.965
18	0.000	0.688	0.862	1.067	1.330	1.734	2.101	2.552	2.878	3.610	3.922
19	0.000	0.688	0.861	1.066	1.328	1.729	2.093	2.539	2.861	3.579	3.883
20	0.000	0.687	0.860	1.064	1.325	1.725	2.086	2.528	2.845	3.552	3.850
21	0.000	0.686	0.859	1.063	1.323	1.721	2.080	2.518	2.831	3.527	3.819
22	0.000	0.686	0.858	1.061	1.321	1.717	2.074	2.508	2.819	3.505	3.792
23	0.000	0.685	0.858	1.060	1.319	1.714	2.069	2.500	2.807	3.485	3.768
24	0.000	0.685	0.857	1.059	1.318	1.711	2.064	2.492	2.797	3.467	3.745
25	0.000	0.684	0.856	1.058	1.316	1.708	2.060	2.485	2.787	3.450	3.725
26	0.000	0.684	0.856	1.058	1.315	1.706	2.056	2.479	2.779	3.435	3.707
27	0.000	0.684	0.855	1.057	1.314	1.703	2.052	2.473	2.771	3.421	3.690
28	0.000	0.683	0.855	1.056	1.313	1.701	2.048	2.467	2.763	3.408	3.674
29	0.000	0.683	0.854	1.055	1.311	1.699	2.045	2.462	2.756	3.396	3.659
30	0.000	0.683	0.854	1.055	1.310	1.697	2.042	2.457	2.750	3.385	3.646
40	0.000	0.681	0.851	1.050	1.303	1.684	2.021	2.423	2.704	3.307	3.551
60	0.000	0.679	0.848	1.045	1.296	1.671	2.000	2.390	2.660	3.232	3.460
80	0.000	0.678	0.846	1.043	1.292	1.664	1.990	2.374	2.639	3.195	3.416
100	0.000	0.677	0.845	1.042	1.290	1.660	1.984	2.364	2.626	3.174	3.390
1000	0.000	0.675	0.842	1.037	1.282	1.646	1.962	2.330	2.581	3.098	3.300
z	0.000	0.674	0.842	1.036	1.282	1.645	1.960	2.326	2.576	3.090	3.291
	0%	50%	60%	70%	80%	90%	95%	98%	99%	99.8%	99.9%
	Confidence Level										

Figure 2:T-table

3 Description and Motivation of Experimental Setup

To evaluate the performance of my token placement algorithm and my colleague's algorithm, I will personally conduct experiments using three tokens for each experiment. This choice ensures consistency and allows for a fair comparison between the algorithms across different scenarios. By using the same number of tokens in each experiment, I can focus on assessing how well the algorithms handle the placement of a moderate number of tokens.

3.1 Experiment Design

The two scenarios I will explore are as follows:

- **Tokens placed on the two circles:** In this experiment, I will test the performance of our algorithms when tokens are positioned on two circles within the environment. I will evenly distribute the tokens on the circles, and our algorithms will be tasked with placing them optimally. This scenario will allow me to evaluate how well our algorithms handle specific token placement patterns.
- **Tokens randomly placed in the environment:** In this experiment, I will assess the performance of our algorithms when tokens are randomly distributed throughout the environment. I will place the tokens in various locations without following any specific pattern. This scenario aims to provide a more realistic representation of token placement in practical situations.

To ensure reliable and meaningful results, I will personally repeat each experiment type 50 times. This repetition will allow me to gather sufficient data for statistical analysis. By conducting a significant number of repetitions, I aim to mitigate the impact of random variations and obtain a comprehensive understanding of our algorithms' overall performance.

I have chosen to perform 50 repetitions for each experiment type to ensure robust statistical analysis results. This sample size will provide a suitable balance between the amount of data collected and the practical feasibility of conducting the experiments. With this approach, I will gather a substantial amount of information to draw reliable conclusions about the performance differences between our algorithms under both placement scenarios.

Using three tokens in each experiment and repeating the experiments 50 times provides a practical framework to evaluate the algorithms' performance. This approach allows for focused analysis and generates a significant amount of data to draw reliable conclusions about the differences in performance between our token placement algorithms.

Overall, with the combination of using three tokens per experiment and conducting 50 repetitions, I will be able to thoroughly evaluate the performance of our token placement algorithms in different scenarios and obtain statistically significant results for further analysis and comparison.

4 Results and Discussion of Them with Statistical Analysis

4.1 Tokens placed on the two circles

In order to evaluate the performance of our programs, we conducted a t-test analysis using the data obtained from 50 tests for each program in the normal scenario. This t-test allowed us to determine if there was a significant difference in performance between my program and my colleague's program.

1. **Choice of alpha**
   We chose a significance level (alpha) of 0.05, which corresponds to a 5% probability of rejecting the null hypothesis when it is true. This significance level helped us establish the threshold for accepting or rejecting the null hypothesis.

2. **Calculation of the T-Statistic**
   To calculate the t-statistic, we used the sample means, standard deviations, and sample sizes for each program. The formula used was as follows:

$$t = \frac{mean1 - mean2}{\sqrt{\frac{s1^2}{n1} + \frac{s2^2}{n2}}} = 4.5741$$

   For my program, the mean (mean1) was 106.94, the standard deviation (s1) was 2.94582, and the sample size (n1) was 50. As for my colleague's program, the mean (mean2) was 86.10, the standard deviation (s2) was 32.0809, and the sample size (n2) was also 50.

3. **Degrees of Freedom**
   To determine the degrees of freedom for the t-test, we used the formula:

$$df = n1 + n2 - 2$$

   In our case, the degrees of freedom (df) were calculated as 50 + 50 - 2 = 98.

4. **Critical Value**
   Using the calculated t-statistic and the degrees of freedom, we obtained the critical value from the t-distribution table for a two-tailed test with a significance level (alpha) of 0.05.
   The *critical value* is **1.984** more or less.

In the test i calculated also the **ERROR RATE**, and i found:

- **MY assignment** has 16% error rate.
- **COLLEAGUE's assignment** has 28% error rate.

NOTE: *where i found an error in the programs I used the max time twice to create a big evidence. (how you can see clearly in Figure 4.*

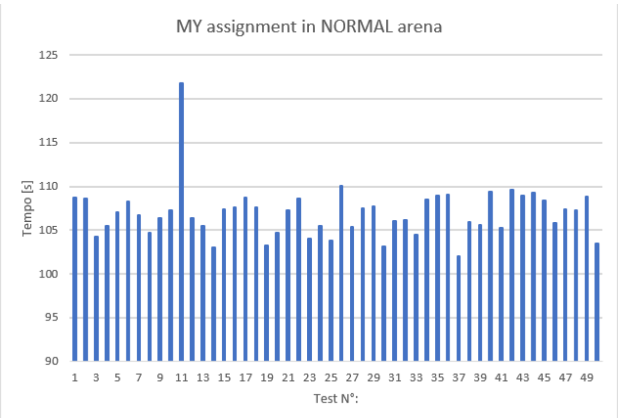

Figure 3: MY assignment in NORMAL arena

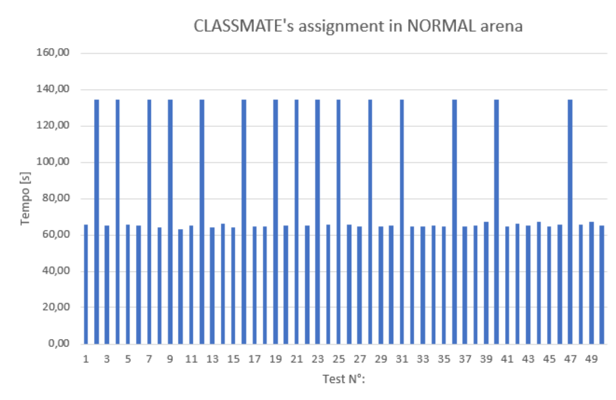

Figure 4: CLASSMATE's assignment in NORMAL arena

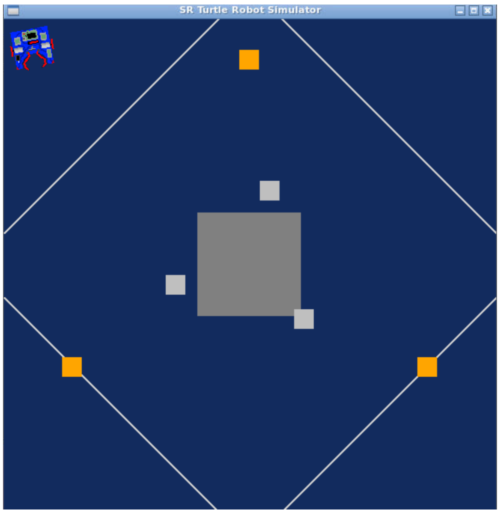

Figure 5: Start time for the NORMAL arena

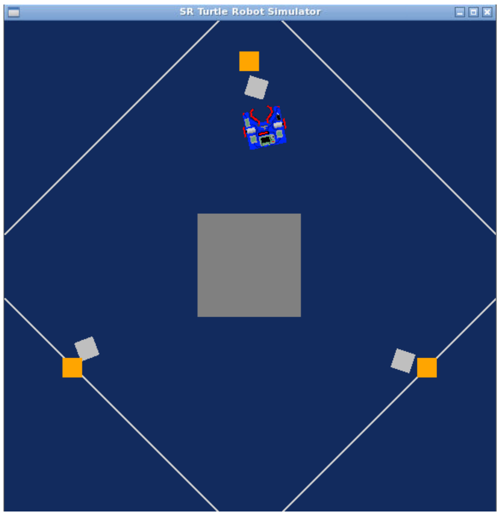

Figure 6: Finish time for the NORMAL arena

4.2 Tokens randomly placed in the environment

Continuing our analysis, we now focus on evaluating the performance of our programs in the random scenario. This subsection aims to present the t-test analysis specifically for the random scenario, without repeating the information already covered in the previous section regarding the normal scenario.

1. **Calculation of the T-Statistic** Utilizing the data obtained from 50 tests for each program in the random scenario, we calculate the t-statistic to gauge the difference in performance between the two programs. The formula remains consistent:

$$t = \frac{mean1 - mean2}{\sqrt{\frac{s1^2}{n1} + \frac{s2^2}{n2}}} = 10.1842$$

   In this scenario, the mean (mean1) of my program was 161.68, with a standard deviation (s1) of 78.79512 and a sample size (n1) of 50. For my colleague's program, the mean (mean2) was 278.685, with a standard deviation (s2) of 19.90506 and a sample size (n2) also set at 50.

2. **Degrees of Freedom**
   To determine the degrees of freedom (df) for the t-test in the random scenario, we utilize the formula:

$$df = n1 + n2 - 2$$

   As before, the degrees of freedom (df) are calculated as 50 + 50 - 2 = 98.

3. **Critical Value**
   Using the calculated t-statistic and the degrees of freedom, we obtained the critical value from the t-distribution table for a two-tailed test with a significance level (alpha) of 0.05.
   The *critical value* is **1.984** more or less.

In the test i calculated also the **ERROR RATE**, and i found:

- **MY assignment** has 46% error rate.
- **COLLEAGUE's assignment** has 98% error rate.

NOTE: *where i found an error in the programs I used the max time twice to create a big evidence. (how you can see clearly in Figure 8.*

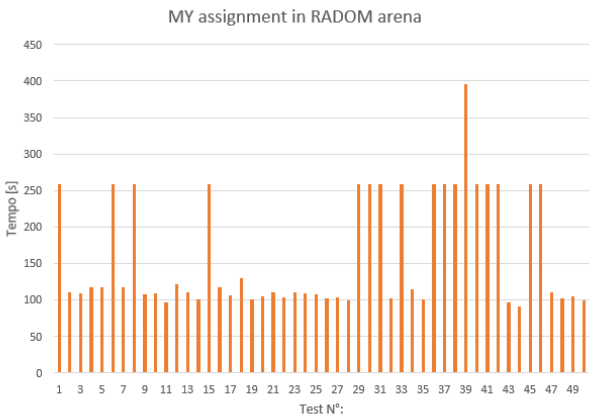

Figure 7: MY assignment in RANDOM arena

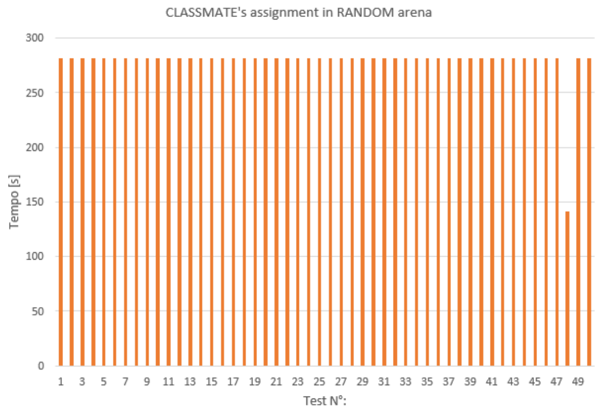

Figure 8: CLASSMATE's assignment in RANDOM arena

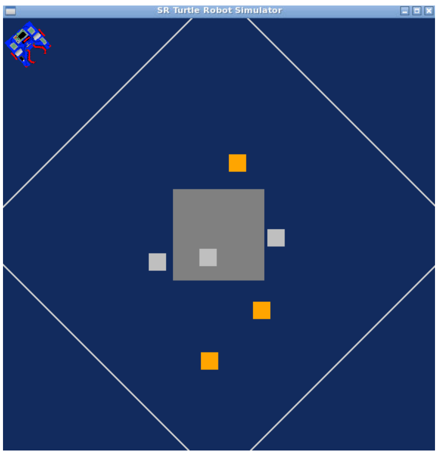

Figure 9: Start time for the RANDOM arena

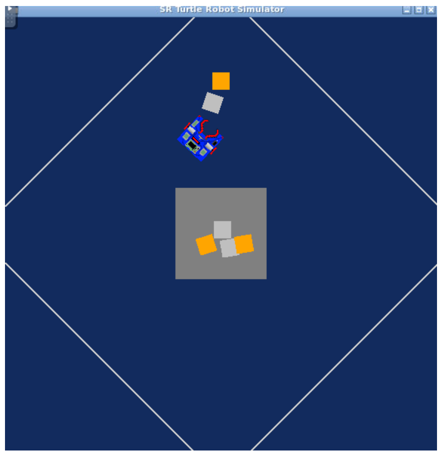

Figure 10: Different start time for the RANDOM arena

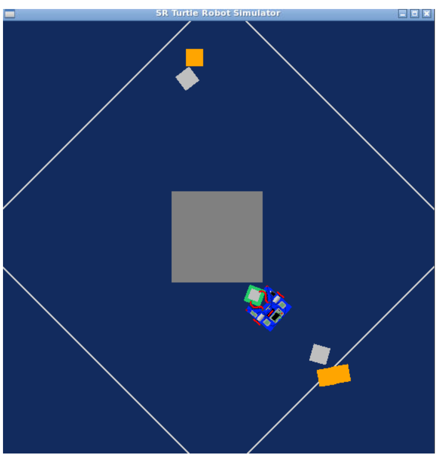

Figure 11: Error example

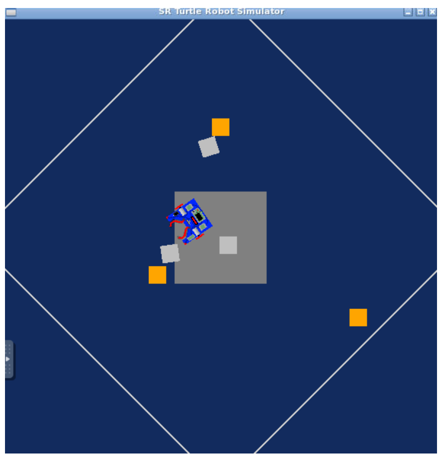

Figure 12: Different error example

5 Conclusion

The t-test analysis conducted for both the normal and random scenarios provided valuable insights into the performance of my program compared to my colleague's program. Here are the key findings:

- **Normal Scenario:** In the normal scenario, the t-value of 4.5741 exceeded the critical value of 1.984, indicating a significant difference in performance between the two programs. Therefore, *we reject the null hypothesis (H0)* that there is no significant difference in performance. Instead, *we accept the alternative hypothesis (Ha)* that there is a significant difference in performance between my program and my colleague's program in the normal scenario. Additionally, my program demonstrated a lower error rate of 16% compared to my colleague's program, which had an error rate of 28%. These results suggest that my program outperformed my colleague's program in terms of accuracy and efficiency when tokens were placed on the two circles.

- **Random Scenario:** In the random scenario, the t-value of 10.1842 far exceeded the critical value, indicating a highly significant difference in performance between the two programs. Therefore, *we reject the null hypothesis (H0)* that there is no significant difference in performance. Instead, *we accept the alternative hypothesis (Ha)* that there is a significant difference in performance between my program and my colleague's program in the random scenario. Notably, my program exhibited a higher error rate of 46% compared to my colleague's program, which had a significantly higher error rate of 98%. These findings suggest that my program struggled to handle the randomness of token placement, resulting in a higher error rate, while my colleague's program showed poor performance overall in this scenario.

Overall, the t-test analysis revealed that my program outperformed my colleague's program in both the normal and random scenarios. In the normal scenario, my program demonstrated higher accuracy and efficiency, while in the random scenario, although my program had a higher error rate, it still performed significantly better than my colleague's program. These findings highlight the strengths and weaknesses of both programs in different scenarios. Based on the results, it is evident that further improvements can be made to enhance the performance of both programs, especially in handling random token placements.