

# Introduction

---

As a developer (*dev*) you are sometimes required to prove your knowledge to prospective clients and employers.

A common way clients assess this for a specific project is to release a *Request for Quotation (RfQ)*. A RfQ can take many forms, but in general it is a questionnaire covering technical and commercial topics. This **Workbook** emulates the technical section of a RfQ.

## Brief

---

The *ACME* Corporation is interested in building a *marketplace* web application (*app*) using *Rails* for one of its product lines. To help it choose the vendor who will undertake the project they have released a **RfQ**. As an aspiring junior *dev* at an up and coming Sydney software startup (*CAX-Dev*) your manager has assigned you to assist with preparation of the RfQ response.

You are assigned (required) to complete all the questions from the *technical* section of the RfQ - which are presented below.

Being a highly valued junior *dev* you are expected to continue working on your other ongoing projects and schedule time for this project accordingly.

*ACME* Corporation is looking for *devs* with an understanding of *Rails*. The following set of questions relate to this RfQ-requirement

### Q1 Describe the architecture of a typical *Rails* application

REF: <https://adrianmejia.com/ruby-on-rails-architectural-design/>

Rails is framework used to develop web applications and is written in the Ruby programming language. The Rails philosophy of 'Convention over Configuration', enables Rails to make the development process faster. These conventions are the strong opinions of Rails that strongly encourage the best way to do many things in web application building, known as "The Rails Way. This not only saves the developer time, but also the need to configure large amounts of code which in turn increases productivity.

The basis architecture of Rails is MVC: Model, View, Controller. These are the 'separation of concerns', where each part has its own responsibility that it handles.

### Model

The model handles the responsibility of interacting with the database and the rules and logic that enable modification of the data. In Ruby on Rails, the models interact with associated elements in the database, representing the corresponding data.

## **View**

The view is the presenting of the data via a rendered webpage, it depicts the user interface of the application. In Ruby on Rails, the view are the HTML files are constructed using embedded Ruby code, and this code is simple with primarily loops and conditionals.

## **Controller**

The controller is the binder that brings together the model and the view. The browser requests are handled by the controller, it gathers the data from the model for processing and sends it to the view for presentation to the user.

**Q2 Identify** a database commonly used in web applications (including *Rails*) and discuss the pros and cons of this database

REF: <https://www.postgresql.org/about/>

REF: <https://www.postgresql.org/docs/current/history.html>

REF: <https://en.wikipedia.org/wiki/ACID>

REF <https://www.guru99.com/introduction-postgresql.html>

REF <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>

## **PostgreSQL**

PostgreSQL was born out of the POSTGRES project in 1986, led by Professor Michael Stonebreaker at the University of California at Berkeley, currently though it is continually developed and maintained by the PostgreSQL Global Development Group, which is a group of dedicated developers.

Although one of the oldest open source database management system, PostgreSQL is one of the most advanced and runs on all major operating systems. This robust database has more than 30 years of active development, which has resulted in PostgreSQL earning a reputation of reliability, integrity and stability. PostgreSQL has many features which are designed to help application building by developers and to help the building of fault-tolerant environments by administrators which enables data integrity.

Key Advantages:

- Open Source- PostgreSQL is a true open source. Offering enterprise grade performance and features for free, allowing the freedom to use, modify and implement it as needed.
- Documentation and support- Due to the vastness of this database and its many years of experience and operation, coming with that is a large amount of documentation. Its dedicated community of supporters regularly contribute

by finding bugs, providing fixes and updating. This ensures this DBMS keeps its reputation of robustness and security.

- Extensibility – highly customizable. For example, personal data types can be defined, customer functions created, and alternative programming languages can be used.
- ACID compliant- i.e. Atomicity, Consistency, Isolation, Durability. This is a set of properties that guarantee validity and reliable processing even in the event of failure. An ACID compliant database management system protects the data despite any failures.
- SQL Compliance- PostgreSQL conforms to 160 out of the 179 features required for full core SQL:2016 standards, which is more than any other DBMS.
- Diversity- PostgreSQL is diverse in many features including its extended functions such as DBlink which can enable database linking and PostGIS for geographic data storage. It also supports a diverse amount of index types which can enable faster data retrieval.
- Geographic objects- PostgreSQL supports geographic objects allowing for location queries in SQL.

#### Key Disadvantages:

- Memory- PostgreSQL can be described as power hungry. When a new client connection is made, PostgreSQL forks a process, which allocates about 10MB of memory for each new process. This can add up very fast especially in larger databases.
- Speed- It is not as fast as other DBMS's such as MySQL, so whilst it excels in optimization it lacks in speed.

ACME Corporation is very big on project management, documentation and process. This will be a key metric in their decision to award the project. The following set of questions relate to this RfQ-requirement.

#### **Q3 Discuss the implementation of Agile project management methodology**

REF <https://www.atlassian.com/agile>

REF [https://en.wikipedia.org/wiki/Agile\\_software\\_development#Overview](https://en.wikipedia.org/wiki/Agile_software_development#Overview)

REF [https://monday.com/blog/project-management/introduction-to-agile/?marketing\\_source=adwordssearch&marketing\\_campaign=au-s-dsa-e-desk-monday&aw\\_keyword=&aw\\_match\\_type=b&qclid=EAlaIQobChMI7rXa-WP6glVIX4rCh0-6QZFEEAAYASAAEgKK\\_vD\\_BwE](https://monday.com/blog/project-management/introduction-to-agile/?marketing_source=adwordssearch&marketing_campaign=au-s-dsa-e-desk-monday&aw_keyword=&aw_match_type=b&qclid=EAlaIQobChMI7rXa-WP6glVIX4rCh0-6QZFEEAAYASAAEgKK_vD_BwE)

REF <https://www.guru99.com/agile-scrum-extreme-testing.html#3>

REF <https://www.atlassian.com/agile/scrum>

REF <https://www.atlassian.com/agile/scrum/roles>

REF [https://en.wikipedia.org/wiki/Scrum\\_\(software\\_development\)](https://en.wikipedia.org/wiki/Scrum_(software_development))

REF <http://agilemanifesto.org/>

## **Agile Manifesto**

*‘Individuals and interactions over processes and tools  
Working software over comprehensive documentation  
Customer collaboration over contract negotiation  
Responding to change over following a plan*

*That is, while there is value in the items on  
the right, we value the items on the left more.’*

## **What is Agile Methodology?**

Described as an iterative approach, Agile project management methodology is the breaking down of larger projects into smaller more manageable increments. These iterations, also called sprints, are consistent time intervals whereby at the end of each something of value is produced for the project. Then whatever is produced is presented forth to stakeholders or users for direct feedback. This means development and testing are parallel.

An example of the most popular Agile framework is Scrum. Described below is a summary of this framework and how it operates.

## **Scrum**

A scrum is described as an iterative framework whereby teams can work together in product development. Usually applied in software development, but it has been used in other team avenues. A Scrum fundamental is the Empirical Process, which highlights that a task cannot be fully understood immediately, rather beginning the work and learning throughout the process is more beneficial. The members that create the basic Scrum team are the Product Owner, the Development Team and the Scrum Master.

### ***Product Owner***

The role of the Product Owner is to represent the business and other stakeholders and is in a sense the be ‘voice of the customer’. This involves having a clear understanding of the customer and create the vision of what the team is delivering to the customer. The product owner takes responsibility for the prioritization of the work and adds them to the Product Backlog and keeping management of it. This role involves excellent communication skills and also negotiation and collaboration skills. It focuses on the business side of product development whereby they control the risks and achieve maximum value of the project.

### ***The Development Team***

The development team are the works who carry out the tasks in the sprints. This team does not solely consist of just engineers but can include all sorts of people that play a part in the development process such as designers and writers. This team needs to be able to be self-organizing and self-manage the workload to deliver the work in the sprint. To ensure the team is on track they meet during a sprint at a daily scrum or standup. This meeting provides an opportunity for members to convey any

obstacles or blockers they might need help with whilst also highlighting any successes.

### ***The Scrum Master***

The role of the Scrum Master is a facilitator of the team. Ensure the scrum framework is followed, and also can also be described as a servant-leader. They help or serve the product owner by helping maintain the backlog which ensures the work is properly communicated and understood. They help the development team by assisting with the self-organization in the team, assist with the determination of a “done” increment, help remove or avoid blockers and also overall coaching.

### ***Overview of Workflow***

- **Sprint-** An iteration of a scrum is a sprint, the length is agreed and decided in advance
- **Spring Planning-** This is held at the beginning of a sprint to examine the scope of work intended for the sprint, select the corresponding product backlog items to put into a prepared sprint backlog and agree on the sprint goal.
- **Daily Scrum-** This meeting checks the daily work and allows the team to check for any blockers.
- **Sprint Review-** Held at the end of the spring, this reviews all complete and incomplete work and presents the completed work to the stakeholders.
- **Sprint Retrospective-** This is for past sprint reflection and allows to identify any needed process improvements.

**Q4 Provide an overview and description of a standard source control workflow**

REF [https://en.wikipedia.org/wiki/Version\\_control](https://en.wikipedia.org/wiki/Version_control)

REF <https://www.atlassian.com/git/tutorials/what-is-version-control>

REF <https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow>

## **Version Control**

Version control is a system that help the management files and any changes made. It keeps a track of any modifications made and allows to revert back to any previous versions in the event of mistakes. An excellent example of this is Git

## **Git**

Git is a free and open source version control system, it is designed for source code management, however it does have the ability to management other sets of files.

## **Git Feature Branch Workflow**

The principle of this workflow is that also feature development takes place on dedicated feature branches instead of solely on the master branch. This system enables developers to work on different features without the core codebase being touched. This idea of encapsulating features in dedicated branches permits pull requests, which are a way to alert others of changes to a branch. This is helpful for team collaboration on each feature branch.

**Overview of this workflow:**

- A central repository houses the Master.
- When a new feature is started, a new branch is created for this feature.
- Each branch should have a clear descriptive name with a clear purpose.
- Each feature branch should also be pushed to the central repository, this means the code can be shared without it interfering with the Master.

**Example Process:**

- Begin on the Master Branch.
- Create a new branch for the new feature.
- Checkout to that branch so you are on it locally.
- Create, edit, stage and commit and changes.
- Push your local changes of the branch to the remote central repository.
- Members of the team can now review your pull request of any changes of this new branch and check the work.
- Finally merge the completed feature into the Master.

Q5 Provide an **overview** and **description** of a standard software testing process (e.g. manual testing)

REF <https://www.guru99.com/automation-testing.html#1>

REF <https://browsee.io/blog/automation-testing-explained/>

## **Automatic Testing**

Automatic testing is a testing process whereby an automatic comparison is done of the actual outcome versus the expected outcome. This way of testing is effective for repetitive, difficult and/or time-consuming test cases. Automated testing tool is proficient to replay the pre-programmed test cases and compare the outcomes and generate reports. Some examples of testing tools are TestComplete and SilkTest. Some of the essential benefits of automated testing are that it can save time and money for the developer and company, it can significantly increase the coverage of the tests and it can increase test reusability and sharing.

### **Process**

- Define Scope of Automation
- Test Tool Selection

- Planning, Design and Development
- Test Execution
- Maintenance

*Having suffered several cyber attacks in the past and resultant remedial audits ACME Corporation takes compliance, security and privacy very seriously. The following set of questions relate to this RfQ-requirement.*

**Q6 Discuss and analyse requirements related to information system security and how they relate to the project**

REF <https://www.computersciencedegreehub.com/faq/what-is-information-systems-security/>

REF <https://stepuptocrypt.blogspot.com/p/blog-page.html>

REF <https://www.proserveit.com/blog/information-security-requirements>

REF [https://en.wikipedia.org/wiki/Information\\_security](https://en.wikipedia.org/wiki/Information_security)

REF <https://www.techopedia.com/definition/24840/information-systems-security-infosec>

REF [https://www.cisco.com/c/en\\_a/products/security/what-is-information-security-infosec.html](https://www.cisco.com/c/en_a/products/security/what-is-information-security-infosec.html)

## **Information Systems Security (INFOSEC)**

INFOSEC is exclusively the processes related to data security

### **Requirements**

**Requirements for a Marketplace Project  
(eg. User authentication and protection and sensitive information)**

unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction of information

defending against DDoS

protecting user X's data from user Y

sign in to access the whole website?

Only sign in for transactions?

**Alternative Options for information security  
(benefits and trade-offs)**

**Summary of Best Security Option**

**Q7 Discuss** common methods of protecting information and data and how you would apply them to the **project**

### **Methods of Data Protection**

**Design for Implementing basic user login authentication for this project**

**Alternative Options for protecting information and data (benefits and trade-offs)**

### **Specific Methods for a Marketplace Project**

***What Methods***

***Why Chosen***

**Q8 Research** what your legal obligations are in relation to handling user data and how they can be met for the **project**

Basic research performed and basic understanding of legal obligations documented

discussion of one way in which obligations can be met for this project (whether or not they are actually implemented)

discussion of alternate approaches to meet obligations and benefits/trade-offs of each

evidence of discussion of findings and agreement on approach to be used to meet legal obligations

*ACME Corporation has specifically requested the app to be based on a relational database. The next set of questions relate to this RfQ-requirement.*

**Q9 Describe** the structural aspects of the relational database model. Your description should include information about the structure in which data is stored and how relations are represented in that structure.

REF <https://www.guru99.com/relational-data-model-dbms.html#4>

REF <https://www.oracle.com/au/database/what-is-a-relational-database/>

REF <https://opentextbc.ca/dbdesign01/chapter/chapter-7-the-relational-data-model/>

REF <https://www.guru99.com/relational-data-model-dbms.html#2>



REF <https://opentextbc.ca/dbdesign01/chapter/chapter-7-the-relational-data-model/#:~:text=A%20domain%20is%20the%20original,%3A%20Married%2C%20Single%2C%20Divorced.>

The relational database model is defined as an instinctive and candid way of data representation using tables. This model allows any file in the database to be connected via these tables which are inter-related relations. Simplicity and efficiency are two key qualities of the relational database model.

Structure:

- Relational Database

The database using the relational model. It is composed of multiple tables whereby these tables hold all the data.

- Relation (Table)

A relation, also called a file or table is the format for which the data is saved along with its encapsulated entities. These entities within the table are a tuple, commonly known as a row, and an attribute, commonly known as a table column. The name of the table is known as the relational variable.

- Attribute (Column)

This is each column in the table and are principle storage units for the table. These attributes define the table values permitted for that column. The total number of attributes in a table is defined as the degree.

- Tuple (Row (unordered))

A tuple is a single row in a table, the data in the rows house single record. This record corresponds to the attributes of the table, and for each record (unique instance of data) in the tuple, it holds its own unique identifying mark which is called a key. The total number rows in the table is defined as the cardinality.

- Relation Schema

This gives a representation of the structure of the data, being the name of the relation along with the names of the attributes within the relation.

- Domain

Is defined as the set of acceptable values an attribute allows for data input. This is grounded on properties and data type assigned to the column.

**Q10 Describe** the integrity aspects of the relational database model. Your description should include information about the types of data integrity and how they can be enforced in a relational database.

REF [https://www.tutorialspoint.com/dbms/relational\\_data\\_model.htm](https://www.tutorialspoint.com/dbms/relational_data_model.htm)  
REF <https://www.smartsheet.com/relational-database-modeling>  
REF <https://www.guru99.com/relational-data-model-dbms.html#2>  
REF <https://www.techopedia.com/definition/7272/foreign-key#:~:text=A%20foreign%20key%20is%20a,establishing%20a%20link%20between%20them.>

The relational integrity constraints of the relation database model consist of key constraints, domain constraints and referential integrity constraints.

- **Key Constraints (entity constraints)**

In all relational databases, there need to be at least one set of attributes (columns) that corresponds to a tuple (row) distinctively. This is called a primary key and every table requires one primary key. The controlling factors of a primary key involved that every row needs to be unique in value (non-identical), and a primary key attribute cannot have a NULL value. It is possible to have multiple attributes with clear unique values, these are referred to as candidate primary keys.

- **Domain Constraints**

These controls are attribute associated. It defines the eligible data valid for input in the columns. These are predefined values. For example, in a date defined attribute, the eligible data for that column is a date, therefore a string would not be acceptable. Domain integrity does permit a null placeholder in an attribute, however this is not representative of a value, it is just a placeholder.

- **Referential integrity constraints**

These controls are in referencing to the usage of foreign keys. Foreign keys are linking attributes (or columns) between two or more tables in a database. It essentially is the linking of two separate tables, from the primary key of one connecting to the foreign key of the other. The integrity of this referencing outlines that in order for cross-table referencing the foreign key in one table, must match the primary key in linked table. If the foreign key is changed in one, the primary key must have been changed in the other. The core referential rule is there must not be any unmatched foreign key values. Additionally, just like the primary key, a foreign key cannot be null.

**Q11 Describe** the manipulative aspects of the relational database model. Your description should include information about the ways in which data is manipulated (added, removed, changed, and retrieved) in a relational database.

REF <https://www.pluralsight.com/guides/sql-data-manipulation-language>  
REF [https://en.wikipedia.org/wiki/Select\\_\(SQL\)](https://en.wikipedia.org/wiki/Select_(SQL))  
REF [https://en.wikipedia.org/wiki/Insert\\_\(SQL\)](https://en.wikipedia.org/wiki/Insert_(SQL))  
REF [https://en.wikipedia.org/wiki/Update\\_\(SQL\)](https://en.wikipedia.org/wiki/Update_(SQL))

REF [https://en.wikipedia.org/wiki/Delete\\_\(SQL\)](https://en.wikipedia.org/wiki/Delete_(SQL))

REF <https://www.tutorialspoint.com/sql/sql-insert-query.htm>

In a relational database, if the data contained is needed to be manipulated this is done using SQL which is Structured Query Language. A sub-language of SQL is data manipulation language (DML). The four most common DML statements for SQL are select, insert, update and delete.

- **SELECT**

The SELECT statement in SQL is used for data retrieval in a singular relations or multiple relations within a database. This statement returns a result set of one or more rows from a table. Additional specificity can be made by using optional phrases along with select, for example; From – Enables specificity of the table selection and Where – Enables specificity of the row selection.

Example: **SELECT \* FROM** (*Table\_name*)

- **INSERT**

The INSERT statement is used for adding singular or multiple records to a table. The basis of an insert statement is INSERT INTO then specifying the columns and values. It is important to note the number of columns in the table you are inserting to, the column name may not need to be specified, but the values in the statement must be in the correct order to match up with the columns they will insert into. Using the additional phrase VALUES enables multi row inserting.

Example: **INSERT INTO** *Table\_name* (**column1** [, **column2**, **column3** ... ])

**INSERT INTO** *Table\_name* (**column-a**, [**column-b**, ...])  
**VALUES** ('value-1a', ['value-1b', ...]),  
('value-2a', ['value-2b', ...])

- **UPDATE**

The UPDATE statement is used for data changing in singular or multiple records in a table. This statement can be utilized by updating all the rows at once or using clauses (conditions) to be more specific and select particular rows. In order for update to be used successfully, it must not conflict with any integrity constraints such a primary key.

Example: **UPDATE** *Table\_name*  
**SET** *column\_name\_1* = 1

- DELETE

The DELETE statement is used for data deletion one or more existing records in a table. When this statement is used alone, it will delete all records from a table, but used along with conditions this enables specificity of the rows for data deletion. An example of this is using the WHERE condition to make a deletion of rows from a table WHERE a column equals a certain value.

Example: **DELETE FROM** *Table\_name*  
**WHERE** *column\_name* = *certain value*;

*The efficiency of an app (i.e. site) and the algorithms used are of the utmost importance. The next set of questions relate to this RfQ-requirement.*

**Q12 Identify and explain** the workings of **TWO sorting** algorithms and **discuss** and **compare** their performance/efficiency (i.e. Big O)

REF <https://www.hackerearth.com/practice/algorithms/sorting/quick-sort/tutorial/>

REF <https://en.wikipedia.org/wiki/Quicksort>

REF [https://www.tutorialspoint.com/data\\_structures\\_algorithms/quick\\_sort\\_algorithm.htm](https://www.tutorialspoint.com/data_structures_algorithms/quick_sort_algorithm.htm)

REF <https://www.geeksforgeeks.org/quick-sort/>

REF <https://www.geeksforgeeks.org/when-does-the-worst-case-of-quicksort-occur/>

REF <https://www.geeksforgeeks.org/quicksort-using-random-pivoting/>

REF <https://www.geeksforgeeks.org/selection-sort/>

REF [https://en.wikipedia.org/wiki/Selection\\_sort](https://en.wikipedia.org/wiki/Selection_sort)

REF [https://www.tutorialspoint.com/data\\_structures\\_algorithms/selection\\_sort\\_algorithm.htm](https://www.tutorialspoint.com/data_structures_algorithms/selection_sort_algorithm.htm)

Sorting algorithms are very important, more specifically, sorting efficiently is key as improves sorting improves the efficiency of other algorithms such as search algorithms that require the input data to be ordered.

### Quick Sort Algorithm

Quick sort is based on the Divide and Conquer method which is breaking the problem down into simpler versions of itself, the basis of this algorithm is in an array, a 'pivot' element is chosen, this element then subdivides the array elements by all the elements that are lesser than the pivot element, and all the elements that are greater than the pivot element and then moves them either side in relation to the pivot. The picking of this pivot element is done in different ways, can be the first element, the last element, a random element or the median element. The important factor is the pivot element itself is the closest middle integer in the array. What follows the aforementioned process is the algorithm then executes recursively the same steps on the new lesser and greater sub-arrays until all sub-arrays contain one element.

The time complexity of this Algorithm for worst case  $O(n^2)$ , which is the larger the input data, the larger the steps. This can be the case when the selected pivot element is either one of the smaller or larger elements in the array, happening when the input data is already sorted or reverse sorted. This scenario can have a lesser change of occurring by choosing the pivot randomly or choosing the middle index. An example below is using random pivoting.

```
partition(arr[], lo, hi)
    pivot = arr[hi]
    i = lo    // place for swapping
    for j := lo to hi - 1 do
        if arr[j] <= pivot then
            swap arr[i] with arr[j]
            i = i + 1
    swap arr[i] with arr[hi]
    return i

partition_r(arr[], lo, hi)
    r = Random Number from lo to hi
    Swap arr[r] and arr[hi]
    return partition(arr, lo, hi)

quicksort(arr[], lo, hi)
    if lo < hi
        p = partition_r(arr, lo, hi)
        quicksort(arr, p-1, hi)
        quicksort(arr, p+1, hi)
```

## Selection Sort Algorithm

Selection Sort Algorithm is a simple algorithm that is an in-place executing comparison algorithm. It executes by dividing an array into two sub-arrays. One sub-array is the sorted array which is at the left end and the other sub-array is the unsorted array at the right end. Initially the sorted array is an empty array and the unsorted array is the whole of the main array. The algorithm then runs through the unsorted array finding the smallest element from it, then putting it at the beginning of

the sorted sub-array. It iterates until all elements are in the sorted array, and each iteration starts at the beginning index of the unsorted array.

Whilst this is a simple style algorithm, it would not be the most efficient particular for large data input. As there are two loops, the worst case complexity is quadratic  $O(n^2)$ , which is the larger the input data, the larger the steps. This can occur if the we are wanting to sort the data in ascending order and the main array is already in descending order. An example of Ruby code for this Algorithm is below.

```
def selection_sort(arr)
  n=arr.length
  for i in 0...n
    min=i
    for j in (i+1)...n
      if arr[j] < arr[min]
        temp = arr[j]
        arr[j] = arr[min]
        arr[min] = temp
      end
    end
  end
  return arr
end

p selection_sort([12,3,1,2,4,70,89,3,3])
```

**Q13 Identify and explain** the workings of **TWO search** algorithms and **discuss** and **compare** their performance/efficiency (i.e. Big O)

REF <https://www.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search>

REF: <https://www.hackerearth.com/practice/algorithms/searching/linear-search/tutorial/>

REF [https://en.wikipedia.org/wiki/Linear\\_search](https://en.wikipedia.org/wiki/Linear_search)

REF <https://medium.com/karuna-sehgal/an-simplified-explanation-of-linear-search-5056942ba965>

REF <https://towardsdatascience.com/a-guide-to-linear-search-and-binary-search-on-arrays-data-structures-algorithms-2c23a74af28a>

## Linear Search

Linear search algorithm is a method used on a collection of items to perform the most basic type of searching algorithm. It is executed by navigating through an entire list from the start to the end, and along the way it individually checks each element to see if a match is found for the target value, or the entire list has been searched.

An example of a Linear search algorithm:

This is a method that checks each value in an array and counts the amount of times that value appears in the array.

Linear search is clearly a basic and simple algorithm; however, this also means it is not the most optimal performing nor efficient algorithm. As shown in the example below, in order for this method to execute, the elements in the array need to be checked individually. This means that as the size of the array of elements grows, so does steps and time taken to check how many times the target is in the array. This results in the time complexity for this algorithm to be  $O(n)$  or linear time, so 100 item array, is 100 steps.

```
def check_target (target)
  arr = [1, 2, 3, 5, 2, 8, 9, 2]
  count=0
  arr.each do |element|
    if element == target
      count +=1
    end
  end
  return count
end

p count_val (2)
```

## Binary Search

Binary search algorithm can be described as an effective Divide and Conquer algorithm for finding a particular item in a sorted array of items. It is executed by associating the target value with the center element in the aforementioned array. If they are equal, the target value is found. However, if they are not equal, and the target value is lesser than the center element, the larger half of the array is discarded. The same applies for if the target value is greater than the center element, the lesser half of the array is discarded. The search then continues to operate on the remaining half, and again repeating this process until the target value is eventually found.

A real-world comparison of a search like this could be using a dictionary to look up a word. Since it is a large book, usually you start with opening the book close to the center. Then if the word you are after is a lower letter or greater letter in the alphabet than the page you turned to, then you would search again in the lower or greater half of the dictionary. Repeating this process until you find the word you are searching. This method is much more efficient, then starting from A and checking every page of the dictionary until you reached the word you are searching. The steps taken doing the first process would be significantly less than the second way of searching the word.

The time complexity of Binary Search is logarithmic time  $O(\log n)$ . Logarithmic growth means it is being divided. So, in the case of Binary Search it is log base 2 of the array collection. Binary Search is highly efficient as the number of steps to the size of the input collection ratio does decrease when the input collection increases.

To summarize, it is clear Binary Search is more efficient in performance time compared to Linear Search as shown in the complexity. Some other notable comparisons are that Binary Search does require the input data to be sorted, whereas Linear does not. Binary Search data access is at random, whilst Linear access' the data sequentially.

```
def binarysearch(arr, i)
  low = 0
  high = arr.length - 1
  while (low < high)
    mid = (low + high) / 2
    if arr[mid] < i
      low = mid + 1
    elsif arr[mid] > i
      high = mid - 1
    else
      return mid
    end
  end
end

puts binarysearch([1,3,12,34,35,46,91,108], 91)
```

*Companies (including ACME Corporation) value previous project experience and case studies. The following set of questions relate to this RfQ requirement.*

- Q14 Conduct **research** into a marketplace website (*app*) and answer the following parts:
- a. List and describe the software used by the *app*.**
  - b. Describe the hardware used to host the *app*.**
  - c. Describe the interaction of technologies within the *app***
  - d. Describe the way data is structured within the *app***
  - e. Identify entities which must be tracked by the *app***



**f. Identify the relationships and associations between the entities you have identified in part (e)**

**g. Design a schema using an Entity Relationship Diagram (ERD) appropriate for the database of this website (assuming a relational database model)**