Hash table: effective algorithm

A Level Advanced
P P P P

Which **three** of these criteria are necessary for an effective hashing algorithm?

- [ ] Must leave no slots empty
- [ ] Must minimise collisions
- [ ] Must always generate the same hash key from the same hash value
- [ ] Must generate a good spread over the problem space

UNIVERSITY OF CAMBRIDGE
Raspberry Pi Foundation

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Hash table: trace algorithm 1

A Level Advanced

C C C C

A hash function is an algorithm that converts a hash key to a hash value. A simple example of a hash function is represented by the pseudocode shown below:

Pseudocode

```
1 FUNCTION hash_it(hash_key, number_of_slots)
2    hash_value = hash_key MOD number_of_slots
3    RETURN hash_value
4 ENDFUNCTION
```

A hash table has 48 slots (buckets) and hash keys are five digit numbers such as those shown in the list below:

- 31971
- 43219
- 56342
- 96756

If the keys listed are hashed using the example of the simple hashing algorithm (shown above), what hash values will be generated?

Key	Hash value
31971	51
43219	67
56342	86
96756	84

Key	Hash value
31971	666
43219	900
56342	1173

96756	2015

Key	Hash value
31971	1
43219	4
56342	8
96756	7

Key	Hash value
31971	3
43219	19
56342	38
96756	36

UNIVERSITY OF CAMBRIDGE
Raspberry Pi Foundation
All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Hash table: trace algorithm 2

A Level Advanced

C C C C

Consider an alphanumeric key that is made up of any five letters or numbers.
An example would be `AZ3FC`.

A hash function calculates hash values as follows:

- each character in the key is converted to its ASCII decimal numeric code (i.e. A will be converted to 65)
- the numeric codes are summed to give a total
- the total is divided by the number of slots in the hash table **using modular division**
- the result is the hash value

If a hash table has 55 slots (buckets) what hash value will be generated for the key value **AZ3FC**?

UNIVERSITY OF CAMBRIDGE
Raspberry Pi Foundation
All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Algorithms: efficiency

GCSE A Level Core
P P P P P P

There are often many different algorithms that can accomplish the same task.

Which of the following characteristics should you consider when comparing the efficiency of two algorithms?

- The number of comparisons
- The number of constants
- The number of subroutines

UNIVERSITY OF CAMBRIDGE
Raspberry Pi Foundation

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Binary search tree: trace

A Level Advanced

C C C C

A wildfowl trust tags the birds that visit over the year with a unique number in order to keep track of them. The details of the birds are stored in a **binary search tree** for efficient searching.

The numbers from the **tags of 13 birds** have been entered into the tree, and these are shown in the diagram in **Figure 1** below.

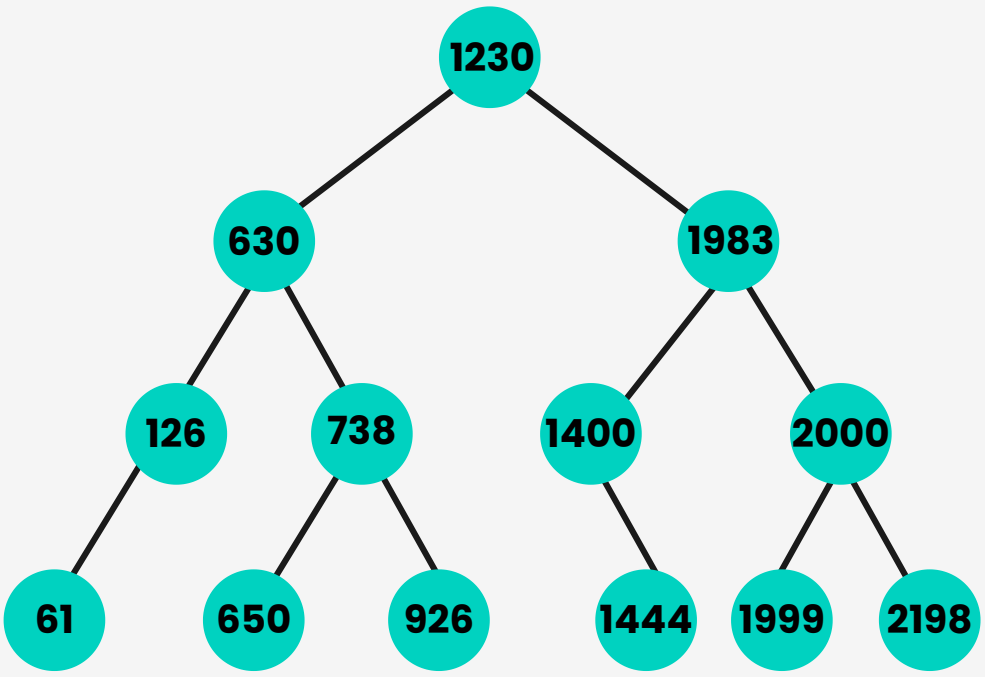

Figure 1: A binary search tree containing data from the tags of 13 birds.

Part A Search for 1999

In what order will the nodes of the binary search tree be visited when the bird with tag number **1999** is searched for?

Select and drag **only the relevant numbers** into the correct order. Not all numbers need to be used.

Available items

- 126
- 630
- 738
- 1230
- 1400
- 1983
- 1999
- 2000

Part B Search for 750

A bird has been found with the tag number **750**. In what order will the nodes of the binary search tree be visited when searching for the tag **750** to find out whether (or not) the tag exists in the binary search tree?

Select and drag **only the relevant numbers** into the correct order. Not all numbers need to be used.

Available items

- 126
- 630
- 650
- 738
- 926
- 1230
- 1400
- 1983
- 2000

Part C Maximum nodes examined

In the example provided, what is the **maximum** number of nodes of the binary search tree that will need to be examined to find any bird?

Part D Time complexity - balanced tree

The tree illustrated in **Figure 1** is an example of a **balanced tree**. For a balanced tree, with n nodes, what is the worst-case time complexity in Big O notation of the binary tree search algorithm?

- $O(log\ n)$
- $O(n\ log\ n)$
- $O(n^2)$
- $O(n)$

UNIVERSITY OF CAMBRIDGE
Raspberry Pi Foundation
All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Big O: order of complexity

Practice 2

P P

Big O notation is a standard way to express an algorithm's time or space complexity. It allows us to understand how the performance of an algorithm scales as the size of the input grows.

Part A

Choose the Big O expression with the **lowest** complexity from the options provided.

- $\mathcal{O}(n!)$
- $\mathcal{O}(n)$
- $\mathcal{O}(\log n)$

Part B

Choose the Big O expression with the **highest** complexity from the options provided.

- $\mathcal{O}(n!)$
- $\mathcal{O}(n)$
- $\mathcal{O}(\log n)$

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Raspberry Pi Foundation

Big O: rank complexities

A Level Advanced

C C C C

Put the following Big O complexities into the order of increasing complexity:

Available items

- O(n^2)
- O(1)
- O(n!)
- O(n log n)
- O(log n)
- O(2^n)
- O(n)

UNIVERSITY OF CAMBRIDGE

Raspberry Pi Foundation

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Big O: determine expression 1

A Level Advanced

C C C C

The number of operations (steps) of an algorithm is given below, where n is the size of the input:

$4n^3 + 3n \log_2 n + 273$

How would the time complexity of this algorithm be expressed in **Big O notation**?

Type your answer in the format $O(n)$ replacing n with the relevant term. If you want to use an exponent, use the ^ symbol, for example **O(n^2)**.

UNIVERSITY OF CAMBRIDGE

Raspberry Pi Foundation

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Big O: determine expression 2

Challenge 2

C C

The number of operations (steps) of an algorithm is given below, where n is the size of the input:

$5n^2 + \log_2 n + 2^n + 18$

How would the time complexity of this algorithm be expressed in **Big O notation**?

Type your answer in the format $\mathcal{O}(n)$, replacing n with the relevant term. If you want to use an exponent, use the ^ symbol, for example, **O(n^2)**.

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.

Raspberry Pi Foundation

Types of complexity

A Level Advanced
P P P P

Fiona needs to write a program that will run in the cloud on Ada Web Services.

Ada Web Services charge money for running code on their servers. The price increases in proportion to how long it takes to run the code on their processors. Customers can make use of as much RAM as they need, for no additional cost.

Select **two** things that Fiona must take into account when designing her algorithm if she wants to minimise costs.

- [ ] Amount of RAM available
- [ ] Whether Ada Web Services uses Big O notation
- [ ] Size of the input
- [ ] Time complexity
- [ ] Space complexity

UNIVERSITY OF CAMBRIDGE
Raspberry Pi Foundation
All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.