COMP10001 Foundations of Computing
Semester 1, 2019

Tutorial Questions: Week 12

— VERSION: 1474, DATE: MAY 23, 2019 —

Discussion

1. What is an “algorithm”? Why are algorithms a large area of Computer Science?
2. What are the two criteria with which we can judge algorithms?
3. What is the difference between exact and approximate approaches to designing an algorithm? Why might an approximate approach be necessary?
4. Identify the following as belonging to the exact or approximate approaches to algorithms, and discuss how they approach solving problems with some examples:
   Brute-Force (Generate and Test), Heuristic Search, Simulation, Divide and Conquer

   Now try Exercise 1
5. What are some of the bases we can store numbers with? How can we convert between bases in Python?
6. Why are binary and hexadecimal convenient for computers? Why is decimal more difficult?

   Now try Exercise 2

Exercises

1. Search the following sorted lists for the number 8, using **(a)** Linear search (Brute-Force approach) and **(b)** Binary search (Divide and Conquer approach)

   Think about the best, worst and average case scenarios of these algorithms. For example, can the best case scenario of a Brute-Force algorithm be faster than running the same task with a more clever algorithm?

(a)	1	2	4	5	8	9	10	12	15	19	21	23	25	
(b)	8	9	11	15	16	17	22	24	27	28	29	32	33	
(c)	2	4	5	6	7	9	11	12	13	15	19	22	25	

2. Convert the binary number 1110101 to hexadecimal by filling in the Xs of the following diagram:

$$\begin{array}{ccc}
 & 1110101_2 & \\
 & \Downarrow & \\
1X1_2 & & 01XX_2 \\
\Downarrow & & \Downarrow \\
X1X1_2 & & 01XX_2 \\
\Downarrow & & \Downarrow \\
X_{16} & & X_{16} \\
 & \Downarrow & \\
 & XX_{16} &
\end{array}$$

Problems

1. Write a Brute-Force algorithm to solve the following problem:

   The length of a ship is an integer. The captain has sons and daughters. His age is greater than the number of his children, but less than 100. How old is the captain, how many children does he have and what is the length of the ship if the product of these numbers is 32118?

2. Implement linear search and binary search in Python. For an extra challenge, write a recursive version of binary search.

3. Write a function which takes a string and returns it as a sequence of binary ASCII values, stored as a string in which each character should take up 8 spaces. Write another function which takes this string of 1s and 0s and converts it back into the original string of text. The `ord()`, `chr()` and `int(_, 2)` functions should be useful.