COMP10001 Foundations of Computing
Semester 1, 2021

Tutorial Questions: Week 11

Discussion

1. What is "recursion"? What makes a function recursive?

2. What are the two parts of a recursive function?

3. In what cases is recursion useful? Where should it be used with caution?

   Now try Exercise 1

4. What is an "algorithm"? Why are algorithms a large area of Computer Science?

5. What are the two criteria with which we can judge algorithms?

6. What is the difference between exact and approximate approaches to designing an algorithm? Why might an approximate approach be necessary?

7. 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 2

Exercises

1. Study the following mysterious functions. For each one, answer the following questions:

   - Which part is the base case?
   - Which part is the recursive case?
   - What does the function do?

   (a)
   ```python
   def mystery(x):
       if len(x) == 1:
           return x[0]
       else:
           y = mystery(x[1:])
           if x[0] > y:
               return x[0]
           else:
               return y
   ```

   (b)
   ```python
   def mistero(x):
       a = len(x)
       if a == 1:
           return x[0]
       else:
           y = mistero(x[a//2:])
           z = mistero(x[:a//2])
           if z > y:
               return z
           else:
               return y
   ```

2. 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

Problems

1. Write a recursive function which takes an integer n and calculates the nth fibonacci number. The 0th fibonacci number is 0, the 1st fibonacci number is 1 and all following fibonacci numbers are defined as the sum of the preceding two fibonacci numbers.

2. 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?

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