

CSE 321 – Introduction to Algorithm Design

Homework 04

Deadline: 16 December 23:55

PLEASE DO NOT FORGET TO READ THE NOTES CAREFULLY

THE NOTES ARE CONTINUOUSLY UPDATED

1. **(20 points)** Your middle school math teacher has given you a list of positive and negative integers and wants from you to find the subarray that has minimum sum. For example, if the input is [1, -4, -7, 5, -13, 9, 23, -1], the subarray that gives the minimum sum is [-4, -7, 5, -13]. Design and implement a divide and conquer algorithm in Python 3 that returns the subarray with minimum sum. Explain your algorithm and do worst case analysis as a comment block at the beginning of your implementation file, `subarray_finder_StudentID.py`. Your code will be tested with many different inputs as follows:

```
inpArr = [1, -4, -7, 5, -13, 9, 23, -1]
msa = min_subarray_finder(inpArr)
print(msa)
#Output: [-4, -7, 5, -13]
print(sum(msa))
#Output: -19
```

2. **(20 points)** Your high school English teacher has given you a list of words (strings). Your mission is to find the longest common postfix of given words. Here is an example:

Your input: ["bash", "trash", "backslash", "flash"]

Your output: "ash"

Your input: ["absorptivity", "circularity", "electricity", "importunity", "humanity"]

Your output: "ity"

Design and implement a divide and conquer algorithm to accomplish this task. Explain your algorithm and do worst case analysis as a comment block at the beginning of your implementation file, `lcp_StudentID.py`. Your code will be tested with many different inputs as follows:

```
inpStrings = ["absorptivity", "circularity", \
              "electricity", "importunity", "humanity"]

lcp = longest_common_postfix(inpStrings)
print(lcp)
#Output: ity
```

3. **(20 points)** As you are a new computer engineering student, you want to have books for classes. You have friend that knows a friend that knows another friend who has \underline{n} of the books you want, and he/she lent them to you in an alphabetically (a to z) sorted form. But you needed \underline{m} more books and you bought them from a bookstore again in an alphabetically (a to z) sorted form. You don't merge and sort them (\underline{m} and \underline{n}) alphabetically because the lent books must not be mixed with your own books. Assuming they are merged and sorted, you want to find the k^{th} (starting from 1) book. Design and implement a decrease and conquer algorithm in Python 3 to accomplish this task in $O(\log \underline{n} + \log \underline{m})$ time (assume that comparing two strings is constant time). Explain your algorithm as a comment block at the beginning of your implementation file, `find_kth_book_1_StudentID.py`. Your code will be tested with many different inputs as follows:

```
m = ["algotihm", "programminglanguages", "systemsprogramming"]
n = ["computergraphics", "cprogramming", "oop"]
book = find_kth_book_1(m,n,4)
print(book)
#Output: oop
book = find_kth_book_1(m,n,6)
print(book)
#Output: systemsprogramming
```

4. **(20 points)** The problem in question 3 can also be solved in $O(\log k)$ time. Design and implement a decrease and conquer algorithm in Python 3 to accomplish this task in $O(\log k)$ time. Explain your algorithm as a comment block at the beginning of your implementation file, `find_kth_book_2_StudentID.py`. Your code will be tested with many different inputs as follows:

```
m = ["algotihm", "programminglanguages", "systemsprogramming"]
n = ["computergraphics", "cprogramming", "oop"]
book = find_kth_book_2(m,n,4)
print(book)
#Output: oop
book = find_kth_book_2(m,n,6)
print(book)
#Output: systemsprogramming
```

5. **(20 points)** In quicksort, both Lomuto's and Hoare's partition schemes can be used. Implement quick sort in Python 3 using both and compare the partition schemes. What are their advantages and disadvantages? Answer this question as a comment block at the beginning of your implementation file, `quickSortsComp_StudentID.py`. Your code will be tested with many different inputs as follows:

```
arr = [15,4,68,24,75,16,42]
qsh = quickSortHoare(arr)
print(qsh)
#Output: [4, 15, 16, 24, 42, 68, 75]
qsl = quickSortLomuto(arr)
print(qsl)
#Output: [4, 15, 16, 24, 42, 68, 75]
```

IMPORTANT NOTES

1. Codes will be written using Python 3 (not Python 2). Do not use any additional python libraries. Use only sys library (import sys).
2. Homework's will be submitted in ZIP format. There is no pdf in this homework. The file hierarchy will be this:
CSE321_HW4_StudentID.zip
 - | subarray_finder_StudentID.py
 - | lcp_StudentID.py
 - | find_kth_book_1_StudentID.py
 - | find_kth_book_2_StudentID.py
 - | quickSortsComp_StudentID.py
3. Use homework question forum on Moodle if you have any questions about homework.
4. Cheating will be punished. Taking any code from internet is also forbidden. (Grade will be -100)
5. No late submissions.