Data Structure Assignment 10

Paper homework

1. Please write down the stability of following sorting algorithm,and then expain why it is stable or unstable(please give an example to explain that)

Algorithm	Stable or Unstable	Reason and Example
Insertion sort	?	?
Quick sort	?	?
Heap sort	?	?

2. Please give the "detailed process" of the following equation derivation (note that answer without any process will be not accepted)

$$\sum_{1\le i\le k} 2^{i-1}(k-i)=\sum_{1\le i\le k-1} 2^{k-i-1} * i \le \mathrm{n} * \sum_{1\le i\le k-1} \frac{i}{2^i} <$$

$$2n = O(n)$$

General information:

- Deadline: **2019/1/3** (Please submit to TA after class.).
- Notice：You won't get any point if you only write the answer,please list your process and reason.
- Late homework will not be accepted.
- Please write on A4 papers, if there is more than one page, staple them together, and write your student id & name on each page.
- Any copies will be scored as zero. Do not plagiarize.

Quicksort analysis

Choosing a good pivot has a significant impact on the performance of quicksort.

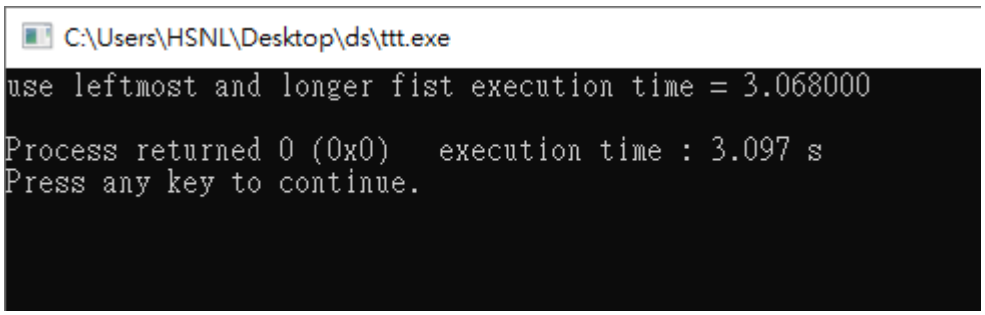
When selecting the pivot , there are two options: leftmost and median of three.

And after swapping the position of the pivot, there are three options for the order of recursion.

Recurse longer sublist first then recurse shorter sublist, or recurse shorter sublist then recurse longer sublist, or recuse the left sublist then recurse the right sublist (like text book p341 program7.6).

In your readme you should show the time for six different combinations of following table (by capturing your program execution time)

Ex:



If you don’t know how to get your program execution time you can follow the program below

```
#include <time.h>

// variable to store your time

clock_t start, end;

double cpu_time_used;

// calculate start time

  start = clock();
```

```
/* your quicksort algorithm put in here*/

end = clock();

   // calculate total cost time

 cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;

 printf("use leftmost and longer fist execution time = %f\n", cpu_time_used);
```

	Longer fisrt	Shorter first	original
leftmost	Time=?	Time=?	Time=?
median of three.	Time=?	Time=?	Time=?

Input.

Test data is in the attachment
You must sort the test data using the 6 methods mentioned above and analyze the time

Output:

Make sure your readme has screenshots of individual execution times for these 6 methods
And analyze why there is such a result

資料結構課程規定

1. 程式執行環境: Windows、Linux。
2. 程式語言: C/C++
3. 程式作業只需提供 source code 和 readme 的說明文件。Source code 只接受.cpp 和.c 檔，其餘檔案類型恕不接受，說明文件請含括您的程式內容的解說，例如，程式執 行流程，程式架構，如何設計功能等，請不要複製題目或複製程式碼註解貼上。
4. 紙本作業請列出推論過程，僅列出答案而未列出過程者，不給分。繳交紙本作 業時，記得寫名字以及學號，請務必記得用 A4 紙張作答，若超過一張，請自行裝 訂起來再交給助教，否則將斟酌扣分。
5. 紙本作業與程式作業請勿抄襲，如有發現一律 0 分計算。
6. 程式作業上傳至 moodle 各章節底下的繳交區。
7. 程式作業與手寫作業皆不接受遲交與補交。程式作業在公布之後的兩個禮拜內 將會開放上傳繳交，請同學們盡早完成作業，避免在最後期限內的一、兩個小時上 傳 moodle 導致發生問題。
8. 每個程式的程式分數佔 80%，說明文件佔 20%，若並未完成作業題目所有要求、 所交程式碼無法執行或執行結果錯誤，將會依照題目要求和執行結果的完程度評 分，其餘的評分項目由當次作業批改的助教來決定。
9. 每次作業壓縮檔(例如 zip, 7z 檔)名稱必須以學號命名，若未註明一律扣該次成績 20 分。

本課程的 TA 時段如下:
星期二（Tues.）p.m. 3:00 - 5:00
星期四（Thu.）p.m. 6:00 - 8:00

若是同學有資料結構相關問題可至高速網路實驗室(資訊系新大樓 5F,65503 室)詢 問，為免同學撲空，如同學無法在助教時間前來，煩請事先與助教預約時間。如有 任何問題請寫信給助教。

助教 何允中 e-mail: p76071323@mail.ncku.edu.tw

潘文傑 e-mail: p76081433@mail.ncku.edu.tw

蘇奕瑋 e-mail: p76081108@mail.ncku.edu.tw

張傑閔 e-mail: p76084499@mail.ncku.edu.tw

蘇雨菲 e-mail: p76075068@mail.ncku.edu.tw

Course Provisions

1. Program execution environment : Windows、Linux
2. Programming language : C/C++ (Languages other than C/C++ are not accepted)
3. Submitted programming homework must include source code in .cpp or .c data type, and readme document. You are required to address the **program architecture**, **program functions and how you design your program** in readme file. Do not just write the pseudo code or even just copy and paste your code!
4. You won't get any point for paper homework if you only write the answers without addressing your process and reasons. **Please do your work on A4 papers. If there is more than one page, please staple them together, and write your student id & name on each page**. Points will be deducted otherwise.
5. **There is a "zero tolerance" for plagiarism. You will receive a score of zero if you get caught plagiarizing.**
6. Please submit your programing homework to moodle.
7. Late homework is not accepted.
8. Programming homework grade is divided into two parts: 80% for the code and 20% for the readme file. **Partial points will still be awarded if the output results of your program are partly correct**. The remaining grading standards are decided by the TAs.
9. **Please name the filename of your submitted compressed file (e.g. zip, 7z) after your** student ID number. 20 points will be deducted otherwise.

TA time of the course:
Tues. 15:00 - 17:00
Thu. 18:00 – 20:00

If you have any question, please come to our lab at TA time (CSIE Bldg. Room 65503). If you are not available to come at TA time, please make another appointment with the TAs. You can also mail us about your questions.

TA 何允中 e-mail: p76071323@mail.ncku.edu.tw

潘文傑 e-mail: p76081433@mail.ncku.edu.tw

蘇奕瑋 e-mail: p76081108@mail.ncku.edu.tw

張傑閔 e-mail: p76084499@mail.ncku.edu.tw

蘇雨菲 e-mail: p76075068@mail.ncku.edu.tw