CISS350: Data Structures and Advanced Algorithms
Quiz q10304

Name: _____YOUR EMAIL_____   Score: _____

All questions below refer to our standard quicksort. (Unless otherwise state, the pivot selection is by median-of-three and the partition algorithm is what I call pLRT.) For big-O, your answer should be of the form $O(n^k \lg^\ell n)$ for $k = 0, 1, 2, ...$ $\ell = 0, 1, 2, ...$ Of course $O(n^0)$ is $O(1)$.

Q1. Suppose you want to perform quicksort on $x = \{5, 3, 1, 4, 6\}$. What is the array after performing the median-of-3 in order to compute the pivot? What is the pivot value?
ANSWER:

$x = \{?, ?, ?, ?, ?, ?\}$, pivot value = ?

Q2. Suppose you want to perform quicksort on $x = \{1, 9, 4, 5, 7, 2, 8\}$. Perform ONE partition on x assuming the pivot is 5.
ANSWER:

$\{?, ?, ?, ?, ?, ?, ?\}$

Q3. The following array has gone through one partitioning: $x = \{3, 1, 4, 5, 7, 9, 8\}$. List the possible value(s) that was used as pivot.
ANSWER:

?, ?, ?, ?

(Erase unused ?)

Q4. Is it stable? If it is, write YES. If it is not, write NO and then write down the simplest array of smallest size and using smallest positive integers > 0 which when traced will show that the sorting is not stable.
ANSWER:

YES. ... or ... NO. $\{?, ?, ?, ?, ?, ?\}$.

Q5. What is the big-O of the runtime in terms of n where n is the size of the array?
ANSWER:

$O(?)$

Q6. What is the big-O of the average runtime in terms of n where n is the size of the array?

Answer:

$O(?)$

Q7. What is the space complexity (i.e., memory usage) in terms of n where n is the size of the array?

Answer:

$O(?)$

Instructions

In the file `thispreamble.tex` look for

```
\renewcommand\AUTHOR{}
```

and enter your email address:

```
\renewcommand\AUTHOR{jdoe5@cougars.ccis.edu}
```

(This is not really necessary since alex will change that for you when you execute `make`.) In your bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`.

Enter your answers in `main.tex`. In the bash shell, execute "`make`" to recompile `main.pdf`. Execute "`make v`" to view `main.pdf`.

For each question, you'll see boxes for you to fill. For small boxes, if you see

```
1 + 1 = \answerbox{}.
```

you do this:

```
1 + 1 = \answerbox{2}.
```

`answerbox` will also appear in "true/false" and "multiple-choice" questions.

For longer answers that need typewriter font, if you see

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
\end{answercode}
```

you do this:

```
Write a C++ statement that declares an integer variable name x.
\begin{answercode}
int x;
\end{answercode}
```

`answercode` will appear in questions asking for code, algorithm, and program output. In this case, indentation and spacing is significant. For program output, I do look at spaces and newlines.

For long answers (not in typewriter font) if you see

```
What is the color of the sky?
\begin{answerlong}
\end{answerlong}
```

you can write

```
What is the color of the sky?
\begin{answerlong}
The color of the sky is blue.
\end{answerlong}
```

A question that begins with "T or F or M" requires you to identify whether it is true or false, or meaningless. "Meaningless" means something's wrong with the question and it is not well-defined. Something like "$1 + 2 = 4$" is either true or false (of course it's false). Something like "$1 +_2 = 4$?" does not make sense.

When writing results of computations, make sure it's simplified. For instance write 2 instead of $1 + 1$.

Higher level classes.

For students beyond 245: You can put LaTeX commands in `answerlong`.

More examples of meaningless statements: Questions such as "Is $42 = 1 +_2$ true or false?" or "Is $42 = \{2\}^{\{3\}}$ true or false?" does not make sense. "Is $P(42) = \{42\}$ true or false?" is meaningless because $P(X)$ is only defined if X is a set. For "Is $1 + 2 + 3$ true or false?", "$1 + 2 + 3$" is well–defined but as a "numerical expression", not as a "proposition", i.e., it cannot be true or false. Therefore "Is $1 + 2 + 3$ true or false?" is also not a well-defined question.

More examples of simplification: When you write down sets, if the answer is $\{1\}$, do not write $\{1, 1\}$. And when the values can be ordered, write the elements of the set in ascending order. When writing polynomials, begin with the highest degree term.

When writing a counterexample, always write the simplest.