Data Structures

Worksheet 3

BITL2

16 September 2021

1. **Warm up:** Answer the following True / False questions.

   (a) If your code initiates a `class` with a constructor, you must delete it with a destructor.

   (b) Without changing any other code, using either of the lines in the function declaration will not produce any errors upon compilation:

   ```
   void func(int this, char that)      void func(int& this, char that)
   ```

   (c) If the body of `try {..}` contains a complation error, then the body of `catch {..}` will be executed.

   (d) Every function $f(n)$ is both $\Omega(1)$ and $O(e^n)$.

   (e) If a function is $O(1)$, then it must be constant.

2. The *time complexity* of an algorithm is $O(f(n))$ if as $n \to \infty$, where n is the size of the input, the algorithm takes at most $M|f(n)|$ time, for some $M \in \mathbf{R}$. You are given Algorithm 1, which is $O(n^a)$, and Algorithm 2, which is $O(n^b)$, for $a, b \in \mathbf{N}$.

   (a) Give another function $f(n) \neq n^a$, so that Algorithm 1 is $O(f(n))$.

   (b) What is the complexity of the algorithm that:

      i. first executes Algorithm 1, then Algorithm 2?

      ii. exeutes Algorithm 1 and Algorithm 2 in parallel?

   (c) Suppose that Algorithm 3 is $O(2^{n+1})$ and Algorithm 4 is $O(2^{2n})$. Are either / both / none of these algorithms $O(2^n)$?

3. Consider the pseudocode on the left, which takes as input a set of numbers $X = \{x_1, \ldots, x_n\}$.

```
1  for $i = n, n-1, \ldots, 2$:
2    for $j = 1, 2, \ldots, i-1$:
3      $x = x_j$
4      if $x > x_{j+1}$:
5        $x_j = x_{j+1}$
6        $x_{j+1} = x$
```

step 0:	1	3	4	2
step 1:				
step 2:				
step 3:				
step 4:				

   (a) How many times is line 3 called?

   (b) What is an upper bound on the number of times line 5 is called?

   (c) In the boxes on the right above, starting with X as given in step 0, write what X looks like every time the order of its elements changes.

   (d) What do you think the code does to X?

4. This question is about the following uncompiled C++ code:

```
 1      #include <iostream>
 2      #include <vector>
 3      #include <string>
 4      using namespace std;
 5
 6      class Player {
 7          \\ Your code here
 8      };
 9
10      bool player_compare(Player &A, Player &B) {
11          \\ Your code here
12      };
13
14      int main() {
15          vector<Player> gg{
16              {"Chad", 10}, {"Brad", 15}, {"Mad", 7}, {"Dad", 20}, {"Tad", 12}
17          };
18          Player& currentwinner = gg[0];
19          for (vector<Player>::iterator it = gg.begin(); it != gg.end(); ++it) {
20              if (player_compare(*it, currentwinner)) {
21                  currentwinner = *it;
22              }
23          }
24          cout << currentwinner.name << endl;
25      }
```

(a) Fill in the mising spots in lines 7 and 11 to make this code print out the name of the player with the largest score.

(b) If line 18 is changed to “`const Player& currentwinner = gg[0];`”, then compilation will produce (at least) two errors. On which lines will these errors occur, and why?

(c) Modify the code so that instead of just the winner being printed, it prints out by how much the winner won, for example, “`Dad wins by 5 points`”.