Practical Task 2.2

(Pass Task)

Submission deadline: Wednesday, March 26
Discussion deadline: Friday, April 11

Instructions

In this task, answer all the following questions and complement each answer with an explanation.

1. Study the Introspective Sort, which is a default sorting algorithm implemented in the Array.Sort method. You must address the following questions:
   - What is special about the Introspective Sort?
   - Does this method result in an unstable sorting; that is, if two elements are seen equal, might their order be not preserved after the data collection is sorted? Or does it perform a stable sorting, which preserves the order of elements that are seen equal?
   - What is the time complexity of this sorting algorithm?
2. In Task 1.1P on the Vector<T> class, you have completed/been provided with a number of methods (and properties), such as Count, Capacity, Add, IndexOf, Insert, Clear, Contains, Remove, and RemoveAt. Answer the following questions:
   - What is the time complexity of each of these operations?
   - Does your implementation match the complexity of the equivalent operations provided by the Microsoft .NET Framework for its List<T> collection?
3. Answer and **explain** whether the following statements are right or wrong.

   3.a. A $\theta(n^3)$ algorithm **always** takes longer to run than a $\theta(\log n)$ algorithm.

   3.b. The best-case time complexity of the Bubble Sort algorithm is always $O(n)$.

   3.c. The worst-case time complexity of the Insertion Sort algorithm is always $O(n^2)$.

   3.d. The Selection Sort is an in-place sorting algorithm.

   3.e. The worst-case space complexity of the Insertion Sort algorithm is $O(1)$.

   3.f. $2 + 2 = O(1)$

   3.g. $n^3 + 10^6 n = O(n^3)$

   3.h. $nlog\, n = O(n)$

   3.i. $log\, n = o(n)$

   3.j. $log\, n + 2^{100} = \Theta(log\, n)$

   3.k. $n\, log\, n = \Omega(n)$

   3.l. $\mathrm{n} + 2^n = O(n)$

   3.m. $n + \frac{100n}{\log n} = o(n)$

   3.n. $\mathrm{n}! = O(n)$

4. Give your best- and worst-case asymptotic runtime analysis to the following code snippets.

   4.1. Let **flag** be a random Boolean variable, whose value is either **true** or **false.**

```
int count = 0;
for (int i = 0; i < n; i++)
{
   if (flag)
   {
       for (j = i+1; i < n; j++)
       {
           count = count + i + j;
       }
   }
}
```

   4.2. Let random() be a function producing a real random number in the range [0,1).

```
int count = 0;
for (int i = 0; i < n; i++)
{
   int num = random();
   if( num < 0.01 )
   {
       count = count + 1;
   }
}
int num = count;
for (int j = 0; j < num; j++)
{
   count = count + j;
}
```

   4.3. Let Compare(x, y) be a method with the time complexity of $\Theta(1)$.

```
for (int i = 0; i < n; i++)
{
   for (int j = 0; j < n-i-1; j++)
   {
       if (a[j] > a[j+1])
       {
           Compare(a[j], a[j + 1]);
       }
   }
}
```

Further Notes

– To get more insights on the Big-O notation and algorithmic complexity, watch the video entitled "Introduction to asymptotic notation: Big O, Big Theta, and Big Omega". This video should help you to answer most of the questions.

– The Asymptotic Cheat Sheet attached as a resource to this task in OnTrack should further clarify the mathematical notation and support your reasoning when dealing with the series of True/False questions.

– You will find ultimate answers to the questions by exploring chapters 4.1.-4.3 of the course book "Data Structures and Algorithms in Java" by Michael T. Goodrich, Irvine Roberto Tamassia, and Michael H. Goldwasser (2014). You may access the book on-line for free from the reading list application in CloudDeakin available in Content $\rightarrow$ Reading List $\rightarrow$ Course Book: Data structures and algorithms in Java.

Submission Instructions and Marking Process

To get your task completed, you must finish the following steps strictly on time.

- **Submit** your answers as a single PDF report via OnTrack submission system.
- Once your answers are accepted by your tutor, you will be invited to **proceed with their discussion through a face-to-face interview**. Specifically, you will need to meet with the tutor to demonstrate and discuss your answers in one of the dedicated practical sessions (run online via MS Teams for online students and on-campus for students who selected to join classes at Burwood\Geelong). Please, come prepared so that the class time is used efficiently and fairly for all students in it. Be on time with respect to the specified discussion deadline.
  You will also need to **answer all additional questions** that your tutor may ask you. Questions will cover the lecture notes; so, attending (or watching) the lectures should help you with this **compulsory** discussion part. You should start the discussion as soon as possible as if your answers are wrong, you may have to pass another round, still before the deadline. Use available attempts properly.

Note that we will not accept your solution after **the submission deadline** and will not discuss it after **the discussion deadline**. If you fail one of the deadlines, you fail the task, and this reduces the chance to pass the unit. Unless extended for all students, the deadlines are strict to guarantee smooth and on-time work throughout the unit.

Remember that this is your responsibility to keep track of your progress in the unit that includes checking which tasks have been marked as completed in the OnTrack system by your marking tutor, and which are still to be finalised. When grading your achievements at the end of the unit, we will solely rely on the records of the OnTrack system and feedback provided by your tutor about your overall progress and the quality of your solutions.