

Algorithm: Algorithms on Titanic Data

1 Description

In this assignment, you will write algorithms to address three questions on Titanic data. Additionally, you need to evaluate your algorithms' goodness.

2 Titanic Data

For this assignment, you are given a file named "TitanicPassenger.txt", which can be downloaded from D2L site. RMS Titanic sank in the North Atlantic the morning of 15 April 1912, after colliding with an iceberg. Of the 1300 passengers aboard, 812 died. (703 of 918 crew members died.) The text file contains a number of rows each of which records a passenger's information including the cabin class (1st, 2nd, and 3rd), age, gender, survived or not (1 survived and 0 died), as well as the passenger's name. Below are the three questions you need to address using algorithms.

1. How many passengers took the 3rd cabin class?
2. How many male passengers are recorded in the given file? How many of them survived?
3. How many female passengers are recorded in the given file? How many of them survived?

For each of the above questions, you need do below:

1. Clearly state what is the algorithm you would like to use in order to approach the question. For example, for the first question, your algorithm should be a sequence of steps to process the file data and calculate the answer to the question. You need to represent your algorithm in pseudo code. (There is no need to code your algorithms using any programming language such as Python. But you are also allowed to code your algorithm using a programming language. Note that each step in the algorithm must be a primitive with a clear definition without ambiguity for a computer (machine or human) to execute. For example, you can assume the step to "open the text file" a primitive in your algorithm so that the file data is ready to be read. You can also assume "Extract cabin-class, age, gender, survived-or-not from the current row being

processed” as a primitive for a human or machine since either human or machine is able to retrieve the information easily.

2. For your algorithm to address the question, you need to explain if your algorithm is able to solve other problems? If yes, what problems and how? Otherwise, why?
3. For your algorithm to address the question, evaluate how good your algorithms is. For example, is your algorithm always able to solve your problem? Is the result reliable? You can also consider some other quality attributes such as efficiency.

3 Turn-in

Turn in your algorithm representation and analysis to D2L dropbox *Algorithms*.