Moldovan Rares-Sebastian

30433

Students' Algorithm teaching tool
Glossary

Version 1.0

Revision History

Date	Version	Description	Author
19/03/2018	1.0	First version of the glossary document.	Rares-Sebastian Moldovan

Table of Contents

Glossary

1. Introduction

The purpose of this document is to introduce the most commonly-used terms in the documentation of the SATT (Students' algorithm teaching tool). The most noteworthy terms are present with their definition so that the interested persons ca use this document as reference when reading the other papers (such as Vision, Use case model etc.)

Most of the terms are not very complicated to understand and might not be necessarily present here for the reader to understand them, but this document makes sure that the reader does not need to do any extra work by finding definitions and explanations to some of the terms involved in describing algorithms and the overall structure of the application.

The document explains terms from the family of algorithms (such as complexity measures), as well as terms present when describing the application (such as types of users, necessary information for accessing the application and some information regarding the structure of an algorithm page). Most importantly, the three types of users are explained with their common characteristics and also the ones that distinguish them from one another.

2. Glossary

Term	Definition and Information	Format	Validation Rules
algorithm	*A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. Algorithms are not necessarily specific only to computer science, but, in the context of the presented application, we refer to computer-related algorithms (such as depth-first search for graphs, sorting algorithms, searching algorithms etc.).*	*An algorithm is most commonly presented as a sequence of pseudocode instructions. However, it can take many forms: actual language-specific code, text description or diagrams.*	*The most important validation rule for an algorithm is the correctness. One cannot discuss about any other feature of an algorithm (such as complexity, completeness or stability) unless it is, first of all, correct).*
complexity	*characterizes the behavior of a system or model whose components interact in multiple ways and follow local rules, meaning there is no reasonable higher instruction to define the various possible interactions*	*The complexity is expressed in the format: O(x), where x is a function of the input data such as "n" or "log(n)".*	*The accepted complexities in computer science are usually polynomial and very seldom for small computational problems, we accept exponential or, in a worst-case scenario, factorial functions for characterizing the complexity.*
optimality	*An algorithm is considered to be optimal when it takes the least sequence of steps possible to reach a solution.*	*Optimality is given by a statement such as "this algorithm is optimal" together with the complexity.*	*Optimality has no degree of comparison (since an algorithm is optimal, there cannot be another which is "more" optimal, or "less" optimal).*

efficiency	*An algorithm is considered to be more efficient than another if its complexity is smaller.*	*Efficiency is proved by proving the complexity measure.*	*Efficiency has a degree of comparison and the most efficient algorithm is the optimal one*
credentials	*They are the input required by the application for the user to log in and access the application.*	*The classical format is a tuple (username, password) where the username can be either a string chosen by the user or seldom the email provided when the registration is being done.*	*They are accepted if and only if the user performed a registration with them and they can be recognized by the system (with access to the application database where all the information is stored regarding the users).*
administrator	*He/she is responsible for the application overall maintenance and for creating new pages.*	*The administrator inherits all characteristics of a normal user and, besides this, it has some special capabilities such as: creating new algorithm pages, assigning regular users as algorithm administrators and creating other administrators.*	*The system validates the user as an administrator whenever he/she logs in in the application (by providing the same information as other users).*
algorithm administrator	*He/she is responsible of a particular algorithm page of the application and has as main responsibilities to contribute to the page and to validate the contributions of other users.*	*The alg. administrator inherits all attributes of a regular user and has more power than them, but less power than the application administrators.*	*The system validates the user as an algorithm administrator and in order for a user to become one, it must pass through the regular user position and be granted the chance to become an alg. administrator by a system administrator.*
regular user	*He/she has no responsibilities regarding the functionality of the application and can be either an observer (study the web page information) or a contributor (submit implementations and explanations).*	*It needs to register in the application server and after that, it has access to the most basic operation of the web page (read).*	*He/she cannot contribute to any of the pages unless it sends the alg. administrator a request and the admin accepts it.*
registration	*The process of registration means providing the necessary information for a user to be recorded in the database. Once a user is registered, it automatically receives the role of a*	*The registration process is performed by the user when he/she wants to have access*	*The registration is successful if and only if the username and email are distinct from the ones already present in the database and if the password respects some minimal constraints*

	regular user.	*to the application and it is performed by providing some basic information such as: name, email and, most importantly, the password which provides access to the application information.*	*(such as the number of characters to be greater or equal to a minimal number, most commonly 6).*
algorithm overview	*A basic presentation of the algorithm, together with the motivation of the appearance and some basic measures (complexity, efficiency etc.).*	*Usually text-description and maybe some diagrams.*	*An algorithm overview is validated by the administrators with respect to the correctness of the information.*
concrete implementation	*The process of providing source code that, through compilation, generates a program that is capable of executing the steps of the abstractly-specified algorithm.*	*Provided in any type of language, from imperative to functional and logic.*	*An implementation is accepted only when it is error-free, maybe warning-free, and correct with respect to the algorithm that it implements.*
reporting tool	*The tool for generating an organized view over a set of data. A relevant example would be a list of users based on their contribution to the implementation.*	*The reporting tool takes as input the list of users and organizes it based on a ranking generated from the number of validated contributions to the algorithm pages.*	*A report is valid if it provides correct point-based information in an afore-mentioned format (best first, increasing/decreasing order).*