



PROGRAMAÇÃO NA PRÁTICA: UMA EXPERIÊNCIA DE ENSINO-APRENDIZAGEM DE ALGORITMOS E ESTRUTURA DE DADOS NA RESOLUÇÃO DE PROBLEMAS DO COTIDIANO APLICADOS À ENGENHARIA ELÉTRICA

Deivison Paulo Alves da Silva – deivison.paulo.eng@gmail.com

João Paulo Marques Tavares – jpmarques.eng.el@gmail.com

Joaquim Henrique Reis – joaquim_mat@hotmail.com

Hugo Xavier Rocha – hxrocha@hotmail.com

Instituto Federal de Educação Ciência e Tecnologia de Goiás (IFG)

Avenida Furnas, Nº 55, Village Imperial.

75524-010 – Itumbiara – Goiás

***Resumo:** A programação computacional compreende um conjunto de instruções que definem um algoritmo e sistematiza a solução de diversos problemas do cotidiano, atribuindo ao computador a tarefa de executá-los e retornar as possíveis soluções. Entretanto, é necessário elaborar o código de uma forma que o computador interprete e execute corretamente, o que ocorre por meio das mais diversas linguagens de programação. O conhecimento das linguagens de programação e suas ferramentas permite ao programador explorar eficientemente os seus recursos na elaboração de um algoritmo. Nesse contexto, foi criado no Instituto Federal de Goiás (IFG) – Campus Itumbiara, um projeto de extensão que visa preparar os estudantes para resolver desafios de programação, apresentados por meio dos problemas de maratona. O projeto mencionado foi criar um grupo de estudos para trabalhar conceitos que vão além da sala de aula e aprimorar habilidades como a criatividade, o raciocínio lógico e o trabalho em equipe. A experiência do grupo, os desafios do projeto, os recursos utilizados e alguns dos resultados obtidos são apresentados neste trabalho.*

***Palavras-chave:** Grupo de Programação, Algoritmos, Maratona, Desafios, Extensão*

1. INTRODUÇÃO

A lógica de programação na engenharia auxilia em aprimorar a habilidade de desenvolvimento de métodos para solução de problemas complexos, os quais se enquadram em diversas situações do cotidiano do profissional de engenharia, ciências da computação e áreas correlatas. A busca por estas soluções ocorre por meios de procedimentos lógicos, denominados algoritmos, traduzidos para linguagem computacional por meio das lógicas de programação (CAETANO, 2014).

As linguagens de programação são metodologias que se caracterizam por um conjunto de instruções de forma que o computador possa interpretar o algoritmo e executá-lo. Nesse contexto, até o próprio conjunto de instruções de um processador pode ser entendido como uma “linguagem de programação”, embora essa linguagem primitiva, também chamada de “linguagem de máquina” não é aplicável para alguns algoritmos, demandando um esforço muito grande na elaboração de programas mais complexos. Sendo assim, foram



desenvolvidas, ao longo da história da computação, diversas “linguagens de programação”, cada qual, a seu tempo, introduzindo facilidades e recursos que foram tornando a tarefa de programar mais fácil e menos susceptível a erros (GUDWIN, 1997).

Com a crescente evolução da informática e das linguagens de computação, as atividades relacionadas a esta área apresentaram um grande impacto, iniciando-se pelas engenharias, onde passou a ter papel de grande importância, sendo utilizadas como técnicas de análise e aquisição de resultados em simulações de problemas complexos, onde busca-se a construção de modelos que representem com fidelidade uma situação real, possibilitando a redução de erros e melhoria no desempenho (OLIVEIRA, 2014).

A Lei nº 11.892/2008 que instituiu a Rede Federal de Educação Profissional, Científica e Tecnológica e criou os Institutos Federais, define em seu Art. 7º Inciso V que um dos objetivos dos Institutos é o de “desenvolver atividades de extensão de acordo com os princípios e finalidades da educação profissional e tecnológica, em articulação com o mundo do trabalho e os segmentos sociais, e com ênfase na produção, desenvolvimento e difusão de conhecimentos científicos e tecnológicos” (BRASIL, 2008).

A partir da temática anteriormente apresentada, o presente artigo trata da criação de um grupo de estudos no Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus Itumbiara, como projeto de extensão, visando a participação dos alunos em competições voltadas à programação computacional. Nesse aspecto, os objetivos, os desafios, os métodos e as experiências técnicas e didáticas oriundas deste projeto são apresentadas ao longo do trabalho.

2. O GRUPO DE PROGRAMAÇÃO

O grupo de programação caracteriza-se como um projeto de extensão do Instituto Federal de Educação, Ciência e Tecnologia de Goiás – Campus Itumbiara e foi composto por alunos do curso de Engenharia Elétrica, sendo que o intuito inicial do projeto era preparar os estudantes para maratonas de programação. Os alunos estavam sob a orientação do professor Hugo Xavier Rocha, que ministra a disciplina de Estrutura de Dados na instituição. O grupo reunia-se em encontros semanais, que comumente ocorriam na biblioteca da instituição em um horário que fosse possível haver um maior número de integrantes que participassem assiduamente no grupo.

2.1. Objetivos

Dentre os objetivos do projeto, destaca-se a preparação dos alunos para maratonas de programação por meio de discussões e resoluções de problemas apresentados em maratonas anteriores; tais problemas são abordados de forma contextualizada e desafiadora, exigindo do aluno uma visão mais crítica e habilidades que envolvem tanto a lógica e conceitos de programação, abordados em sala, quanto conhecimentos interdisciplinares. O desenvolvimento de habilidades necessárias para se executar o trabalho em equipe é outro objetivo a ser destacado, uma vez que o ambiente de maratona privilegia a construção do conhecimento pela interação do grupo, onde todos trabalham juntos na resolução do problema.

Apesar do foco principal do projeto ser voltado para preparação dos integrantes para maratonas de programação, um dos intuítos do projeto era formar um grupo de estudos, de modo que os integrantes pudessem revisar conteúdos apresentados anteriormente e estudar



outros tópicos avançados relacionados à Lógica e Linguagem de Programação, como Inteligência Artificial.

2.2. A maratona de programação

A maratona de programação é um evento da Sociedade Brasileira de Computação (SBC), em parceria com a Fundação Carlos Chagas, que promovem competições regionais classificatórias para as finais mundiais do concurso de programação da ACM - *International Collegiate Programming Contest*, e é integrante da regional sul-americana. Destina-se a alunos dos cursos de graduação e de início de pós graduação na área de computação e correlatas, como a Engenharia Elétrica (IME-USP, 2014).

A competição busca em seus participantes o trabalho em equipe, assim como habilidades de análise, compreensão e resolução de problemas complexos adicionados a um ambiente, que mesmo sob pressão, requer criatividade, domínio da lógica e de linguagens de programação complexas, além de colocar à prova todo o conhecimento adquirido na preparação. Por essa razão, a cada realização do evento, cresce o interesse do meio corporativo, que atenta-se aos competidores e em seu desempenho nas maratonas, pois os resultados na competição podem ser indicadores de que aqueles que se sobressaem perante os problemas propostos estão capacitados a lidar com problemas similares do cotidiano do mercado de trabalho.

Na maratona, os times são compostos de até quatro alunos, sendo um reserva, aos quais são encarregados a tarefa de resolver a maior quantidade de problemas apresentados no dia da competição, em menos tempo e com o menor número de tentativas falhas. Por essa razão, a competição requer muita preparação e estudo e é nesse aspecto que ela torna-se conhecida e desafiadora, trazendo assim, reconhecimento ao esforço dos times que se sobressaem.

2.3. Desafios

O grupo de programação, por lidar com problemas mais desafiadores, configura-se como uma prática de extensão melhor trabalhada e que teria melhor rendimento entre alunos que já concluíram da disciplina de Estrutura de Dados, ministrada no segundo período de engenharia elétrica, na instituição. Com isso, o subgrupo de alunos concluintes desta disciplina representou um fator limitante e um dos desafios do projeto. Além disso, a estrutura do curso não é focada para computação, fator este que contribui como elemento de dispersão dos estudantes e dificultam o interesse deles em fazer parte do projeto. Outros fatores como a complexidade dos problemas propostos, onde se utiliza de ferramentas computacionais e estruturas lógicas elaboradas para resolução da maioria dos problemas, e a indisponibilidade, em termos de horário, de muitos alunos, também contribuíram para que tal prática não fosse amplamente difundida na instituição.

2.4. Histórico das disciplinas de Lógica e Linguagem de Programação na grade curricular

O estudo de disciplinas que envolvem programação inicia-se no curso de Engenharia Elétrica da instituição pela disciplina de Algoritmos e Linguagens de Programação, no 1º período do curso. Nessa disciplina, são abordados conceitos básicos da lógica de programação, com enfoque nas linguagens C e C++, as quais são essencialmente empregadas na resolução dos problemas de maratona. O domínio da lógica e a capacidade de resolução de



problemas que envolvem programação são as principais habilidades adquiridas ao término da disciplina.

Na evolução dessa vertente do curso, a disciplina de Estrutura de Dados, ministrada no 2º período, alia os conhecimentos previamente adquiridos de lógica de programação com uma visão mais elaborada dos códigos, lidando com recursos, métodos e ferramentas computacionais que são empregadas numa vasta gama de problemas práticos, de forma a resolvê-los ou otimizar as soluções já existentes, seja em termos de esforço computacional, alocação de memória ou simplificação do algoritmo.

Após a conclusão da disciplina de Estrutura de Dados, há várias disciplinas que envolvem estudos em programação em sua essência ou de forma parcial, porém com enfoque em outras linguagens de programação. Dentre estas, destaca-se, a disciplina de Informática Aplicada, ministrada no 3º período do curso, com enfoque em programação lógico-matemática, e de modelagem e simulação; Tópicos de Inteligência Artificial, no 7º período, com enfoque em programação lógica e estrutural destinada à aplicação em Redes Neurais Artificiais, em seu treinamento e utilização para reconhecimento de padrões; Eletrônica Digital 2 e Microcontroladores, respectivamente no 4º e no 7º períodos do curso, cujo enfoque é a programação em *hardware* para a montagem de circuitos eletrônicos.

2.5. Preparação para a maratona: SPOJ

O SPOJ (*S*phere *O*nline *J*udge) é uma plataforma online, cujos servidores analisam computacionalmente os programas submetidos pelos seus usuários. Nela, há um suporte para mais de 45 linguagens de programação, com compiladores em linguagens como, por exemplo, C, C++, JAVA, dentre outras. O SPOJ está disponível em vários países, incluindo uma versão em português, no Brasil (GUIMARÃES, 2014).

No SPOJ, um sistema de correção julga se o código enviado pelo usuário está correto quanto à sua sintaxe e estruturação segundo a linguagem selecionada, à alocação de memória, ao tempo de execução e, sobretudo, se as saídas estão fielmente dentro do esperado, para todos os casos de testes contidos num banco de dados do servidor. Caso todos estes parâmetros, que variam conforme o problema, estejam dentro do esperado, o programa é aceito.

Os problemas disponibilizados no SPOJ são oriundos de maratonas de programação e competições correlatas, como olimpíadas de informática a nível regional e nacional, cuja complexidade é variável, mas que tipicamente estão contextualizados a uma situação prática e reforçam a aplicabilidade de conceitos teóricos em seus enunciados, requerendo interpretação e criatividade em sua resolução.

2.6. Aplicação: O problema da transmissão de energia

Dentre os problemas disponíveis no site do SPOJ, foi selecionado para análise um problema relacionado à Engenharia Elétrica, denominado de *Transmissão de Energia*. Em resumo, o objetivo é verificar se todas as subestações de um sistema elétrico encontram-se interligadas ou não, isto é, para cada par de subestações do sistema, verificar se existe pelo menos um caminho que interligue estas duas subestações (GUIMARÃES, 2014). De uma forma simplificada, o elemento chave (o Sistema Elétrico) desse problema pode ser representado por um grafo, deste modo, pode se utilizar de conceitos apresentados na Teoria dos Grafos juntamente com algoritmos computacionais específicos para manipulação de grafos para resolução do problema.

Conforme o enunciado do problema, seriam informados, na primeira linha da entrada, o número de subestações que compõem o sistema (E) e o número de ligações (L), as subestações são indicadas por números que variam de 1 até o número de subestações, tendo fator limitante de 100 subestações. Nas sucessivas (L) linhas de entrada, são informados dois números que representam duas subestações distintas, indicando assim que há uma ligação entre elas.

Inicialmente, com os dados inseridos pelo usuário, buscou-se aplicar conceitos da Teoria dos Grafos previamente estudados, convertendo esses dados em uma matriz correspondente que representasse as relações das subestações do sistema de transmissão, tal matriz é denominada de matriz de adjacência. Matematicamente, a matriz de adjacência $A(G)$ de um grafo $G=G(E, L)$, sendo $E=\{e_1, \dots, e_n\}$ um conjunto dos n vértices e L é o conjunto das arestas (ligações) de G , é uma matriz quadrada de n -ésima ordem, em que os elementos estão distribuídos conforme a equação (1), que atribui 1 às posições na matriz que correspondem a vértices adjacentes e 0 caso contrário (FRITSCHER, 2011).

$$a_{ij} = \begin{cases} 1, & \text{se } (e_i, e_j) \in L \\ 0, & \text{se } (e_i, e_j) \notin L \end{cases} \quad (1)$$

$$(e_0 \ e_f) \rightarrow \begin{array}{c} 1 \\ \vdots \\ e_0 \\ \vdots \\ e_f \\ \vdots \\ E \end{array} \begin{array}{c} \left[\begin{array}{ccccccc} 0 & \dots & 0 & & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & & \boxed{1} & \dots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \dots & \boxed{1} & & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & & 0 & \dots & 0 \end{array} \right] \\ \hline 1 \quad \dots \quad e_0 \quad \dots \quad e_f \quad \dots \quad E \end{array}$$

Figura 1 – Montagem da matriz de adjacência a partir dos dados inseridos pelo usuário.

A figura 1 mostra como os dados de entrada são tratados dentro da matriz adjacência. A existência de uma ligação entre as subestações é indicada na matriz de adjacência, fazendo os elementos da matriz nas posições correspondentes em linha e coluna iguais a 1. Na diagonal principal, considera-se a ligação do ponto com ele mesmo, sendo convencionado 0 o valor de tais elementos para a solução deste problema.

O grafo que representa o sistema elétrico, neste caso, é denominado de grafo não direcionado, assim, a matriz adjacência possui a propriedade de simetria em relação à diagonal principal, isto é, se a está ligado com b , a recíproca também é válida.

Após a montagem da matriz de adjacências, que define as conexões do grafo ou da rede, é necessário verificar se o grafo é conexo ou desconexo e é nessa análise que a matriz de adjacências constitui uma ferramenta de análise dinâmica da rede.

O primeiro passo é atribuir uma condição inicial, isto é, um ponto para iniciar o percurso do grafo. Considerou-se a primeira linha da matriz de adjacências, que contém todas as conexões do elemento 1. Para isso, criou-se uma sub-rotina que registra novas conexões, isto é, conexões com elementos que ainda não foram registrados, e analisa estas conexões nas iterações futuras. Se o elemento não possuir novas conexões, o algoritmo tende a encerrar um

laço que controla a execução da sub-rotina que registra as novas conexões. Após a conclusão do laço, é feita a análise das conexões registradas. Se forem registradas conexões em todos os pontos partindo de um único ponto, a rede é conexa e o programa retorna a resposta "normal". Caso contrário, o programa retorna a mensagem "falha", indicando que nem todos os pontos estão interligados entre si.

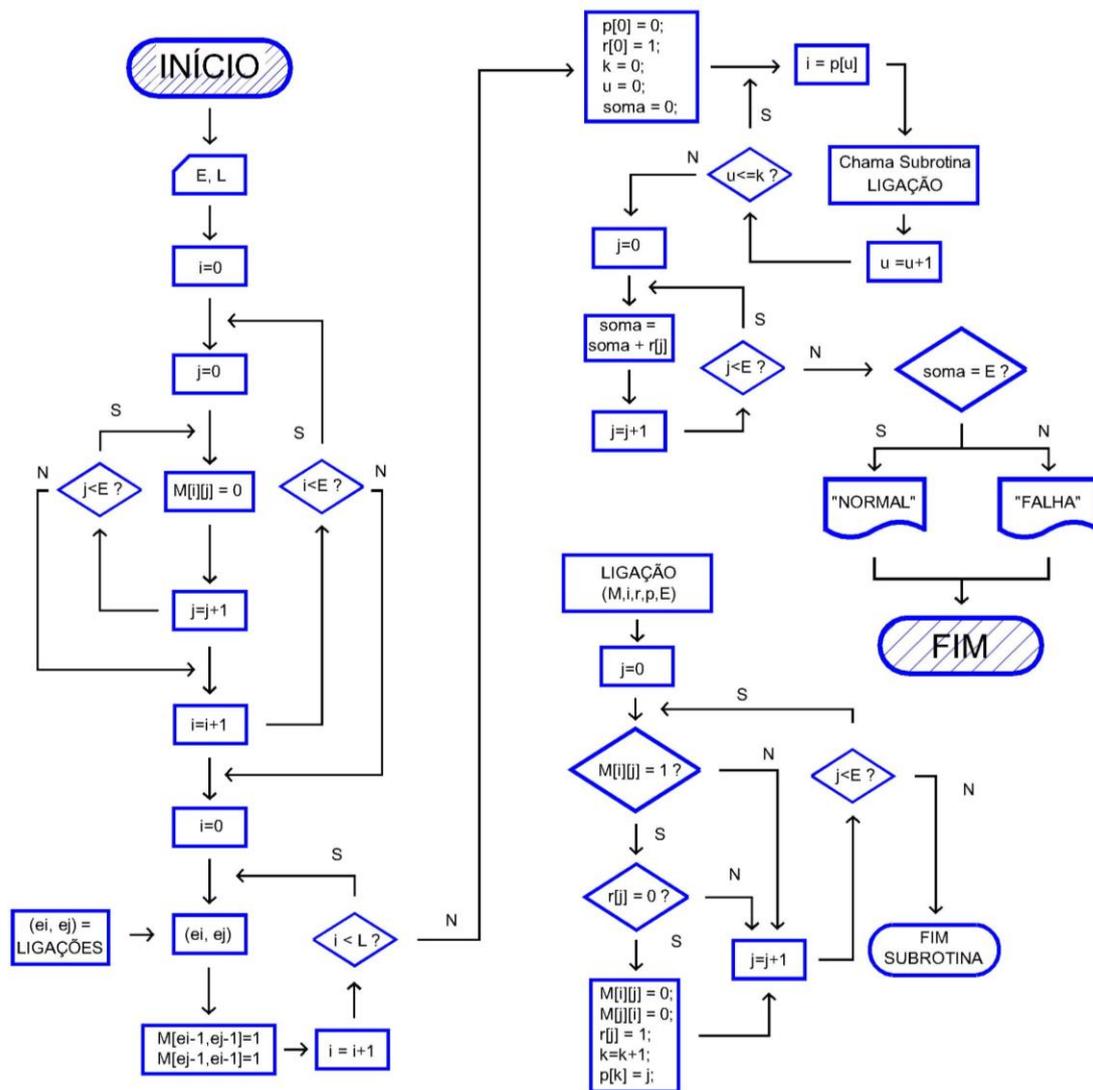


Figura 2 – Fluxograma do algoritmo elaborado para o problema de transmissão de energia.

A figura 2 mostra o fluxograma que representa o algoritmo, em que são esquematizadas três etapas do programa. À esquerda, é mostrado a entrada de dados e a conversão dos dados inseridos pelo usuário na matriz de adjacências. À direita, é mostrado a rotina de busca do grafo, na parte superior, ocorre o gerenciamento da sub-rotina, assim como a coleta e interpretação de seus resultados e, logo abaixo, é apresentado o fluxograma da sub-rotina, assim como os critérios de sua execução.



O algoritmo utilizado caracteriza uma busca em amplitude (*Breadth First Search - BFS*), pois ele registra todas as conexões de um elemento e as registra antes de continuar a execução do programa. Um algoritmo de busca em profundidade (*Depth First Search - DFS*) encerraria o laço na primeira conexão nova registrada e avaliaria as conexões desta, este tipo de busca é melhor aplicado em grafos que descrevem um modelo predominantemente radial, com poucas conexões partindo dos elementos. Para aplicação em uma entrada mais genérica, optou-se por utilizar a busca em amplitude (DROZDEK, 2005).

Dentre os conceitos de programação adquiridos com o grupo, alguns deles foram apresentados anteriormente, pela teoria de grafos. Entretanto, problemas mais complexos requerem o uso de ferramentas computacionais denominadas metaheurísticas, como algoritmos genéticos, por exemplo, que exploram as soluções em sua aleatoriedade e, com sucessivas iterações, a solução tende a convergir para uma solução satisfatória, que pode não ser a melhor solução do problema. O uso dessas ferramentas é viável em determinadas aplicações nas quais o esforço computacional requerido pelo algoritmo inviabiliza a utilização de um método determinístico, isto é, que retorna uma resposta exata (LUCAS, 2002).

3. CONSIDERAÇÕES FINAIS

Nesse aspecto, o conhecimento dos recursos computacionais e sua aplicabilidade é um fator determinante na existência do grupo de programação e os desafios propostos com os problemas de maratona representam o meio ideal para ampliar essa experiência didática e trazer benefícios no que se refere à criatividade, ao raciocínio lógico, ao trabalho em equipe e à aquisição de conhecimentos relacionados à programação computacional. Com isso, espera-se que estes fatores motivem cada vez mais alunos a fazerem parte do grupo e compartilhar essa experiência.

Agradecimentos

Agradecemos a todos aqueles que dedicaram seu tempo em resolver problemas de maratona e contribuíram com o grupo de programação, e também às empresas que organizaram a maratona e tornaram possível essa competição e todos os benefícios didáticos oriundos dela.

4. REFERÊNCIAS BIBLIOGRÁFICAS

BRASIL. Senado Federal, Lei Federal n.º 11.892, de 29 de dezembro de 2008.

CAETANO, D. **Lógica de Programação Para Engenharia: Introdução à Disciplina.** Disponível em: <http://www.caetano.eng.br/aulas/2011b/lpe/lpe_ap01.pdf> Acesso em: 23 mai. 2014.

DROZDEK, Adam. Estruturas de dados e algoritmos em C++. Ed. Thomson Learning, São Paulo - SP, 2005.

FRITSCHER, Eliseu. Propriedades espectrais de um grafo. Universidade Federal do Rio Grande do Sul. Instituto de Matemática. Apostila, 2011.



GUDWIN, Ricardo R. Linguagens de Programação: Notas de aula. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. Apostila, 1997.

GUIMARÃES, W. *et al.* **O que é o SPOJ Brasil?**. Disponível em: <<http://br.spoj.com/embed/info/>> Acesso em: 23 mai. 2014.

GUIMARÃES, W. *et al.* **SPOJ Problem (obi): Transmissão de Energia**. Disponível em: <<http://br.spoj.com/problems/ENERGIA/>> Acesso em: 23 mai. 2014.

IME - USP. **Maratona de Programação: O que é?**. Disponível em: <<http://maratona.ime.usp.br/info14.html>> Acesso em: 23 mai. 2014.

LUCAS, Diogo, C. Algoritmos Genéticos: uma Introdução. Universidade Federal do Rio Grande do Sul. Apostila, 2002.

OLIVEIRA, L. A. G. de. **Introdução à Informática: Histórico e Evolução**. Disponível em: <http://www.dca.ufrn.br/~lmarcos/courses/DCA800/pdf/Apresentacao_historico.pdf> Acesso em: 23 mai. 2014.

COMPUTATION PROGRAMMING PRACTICE: AN EXPERIENCE OF ALGORITHMS AND DATA STRUCTURE LEARNING IN DAILY PROBLEM RESOLUTIONS APPLIED TO ELECTRICAL ENGINEERING.

Abstract: *The computational programming involves a rule set or algorithm and it describes by codes the solution of several practice problems. The computer compiles the code and it returns the outputs. On the other hand, it is necessary that the programmer elaborates a code so that the computer run the code and return the outputs correctly. The programming languages and their resources can be very useful if the programmer knows about them. Hence, it was created on Goiás Federal Institute (IFG) – Campus Itumbiara (Brazil), an extracurricular project whose objective is to solve the programming problems proposed on previous competitions (marathons). The project main idea it is based in a local group of students to training extracurricular topics about computational programming. The improvement of creativity, logical reasoning, and group relationship are some abilities developed by this project. The experiences, challenges, methods used by the group as well as some results are described along this paper.*

Key-words: *programming group, algorithm, competition, challenges, extracurricular project*