

Desarrollo e implementación de un programa de trabajo para el Semillero de Programación

A. Echavarría Uribe
Ingeniería Matemática
Universidad EAFIT
Medellín, Colombia
Email: aechava3@eafit.edu.co

J. F. Cardona Mc'Cormick
Escuela de Ingeniería
Universidad EAFIT
Medellín, Colombia
Email: fcardona@eafit.edu.co

Resumen—

Palabras claves—

I. INTRODUCCIÓN

El Semillero de Programación es un grupo de la Universidad EAFIT en el que los estudiantes con interés en la programación, las matemáticas y los algoritmos tienen un espacio para aprender acerca de estos temas y prepararse para participar en las maratones de programación realizadas por ACIS/REDIS [1] y por la ACM-ICPC [2]. En este semillero se enseñan los temas más útiles [3]–[6] para estas competencias: los algoritmos de grafos, strings y teoría de números, programación dinámica, la recursividad y las estructuras de datos.

Durante los últimos años el Semillero ha estado a cargo de estudiantes destacados en las maratones de programación bajo la supervisión de docentes del Departamento de Ingeniería de Sistemas y, a pesar de que se han trabajado y discutido algoritmos, temas y problemas, no se desarrolló un plan de trabajo para este grupo. Dos consecuencias de lo anterior son que los miembros del Semillero a veces no tenían la fundamentación necesaria para aprender algunos de los algoritmos más complejos y que cuando algún estudiante nuevo se encargaba del Semillero no sabía cuáles eran los conocimientos que los estudiantes tenían y qué temas nuevos por discutir se ajustaban a su nivel. Por lo anterior se decidió crear, desarrollar e implementar un plan de trabajo para el Semillero de modo que este tuviera una estructura de trabajo que permita la apropiación progresiva del conocimiento necesario para poder resolver los problemas de las maratones de programación y que proporcione a quienes se encargarán del Semillero en versiones futuras información acerca de los conocimientos que tienen los estudiantes.

Actualmente, los estudiantes que pertenecen al Semillero de Programación son en su mayoría de tercer semestre, lo que quiere decir que tienen conocimientos acerca de cómo programar mas no conocen las técnicas más utilizadas en la solución de problemas de maratones de programación como los son los algoritmos de grafos, strings, la programación dinámica y los conceptos y algoritmos básicos de teoría de números. El programa desarrollado busca ajustarse a este nivel para poder proporcionar a los estudiantes la fundamentación teórica necesaria para resolver los problemas más comunes presentados en las maratones de programación. En el programa de trabajo se desarrolló documentación, diapositivas y competencias que

servirán como material de trabajo para futuras generaciones del semillero.

Se espera que los temas enseñados a los estudiantes durante este semestre y el próximo sirvan para que ellos tengan un buen desempeño en la Maratón Nacional de Programación ACIS/REDIS que se realiza en octubre y tengan la posibilidad de participar en la Maratón Regional Suramericana ACM-ICPC y en la Maratón Mundial ACM-ICPC bajo el nombre la Universidad.

II. METODOLOGÍA

El plan de trabajo del Semillero está basado principalmente en la metodología utilizada por Steven Halim en el curso Competitive Programming de la Universidad Nacional de Singapur (NUS). Este curso, al igual que el Semillero, busca preparar a los estudiantes para las competencias de la ACM-ICPC y está incorporado como curso electivo para estudiantes de matemáticas, ciencias de la computación e ingeniería electrónica de tercer año con conocimientos previos de programación, algoritmos y estructuras de datos. En el curso se trabajan algoritmos de nivel medio y avanzado pero, a diferencia de un curso de algoritmos corriente, se hace énfasis cómo implementarlos eficientemente y en sus aplicaciones en las competencias de programación, en lugar de enfocarse en las demostraciones de su corrección y el análisis de su complejidad [7].

El método de enseñanza del curso de la NUS consiste tener clases teóricas en las cuales se enseñen los algoritmos a trabajar y hacer las evaluaciones por medio de competencias en las que los problemas, tomados de archivos históricos de competencias anteriores de la ICPC, tengan relación con los temas enseñados. Las dos razones principales por la que decidieron trabajar con competencias en lugar de exámenes son preparar a los estudiantes para las competencias oficiales de la ACM-ICPC y motivarlos a ser cada vez mejores al ponerlos a medir sus habilidades contra los demás estudiantes. Este método ha resultado beneficioso para los estudiantes de la NUS y el Profesor Halim cree que es porque incita a los estudiantes a estudiar más para mejorar ya que buscar ser los mejores es es una característica natural de los humanos.

La principal ventaja que ha tenido este curso en la NUS ha sido darle a los estudiantes talentosos e interesados en la programación un espacio en el que puedan prepararse para las competencias y conocer otras personas interesadas en este tema con las que puedan discutir, competir y formar equipos

que tengan buen desempeño en las competencias de la ACM-ICPC, llegando a participar en la Maratón Mundial ACM-ICPC [8]. El Semillero de Programación de EAFIT tiene el mismo objetivo y es por esto que se decidió desarrollar un plan de trabajo basado en esta metodología y adaptarlo para el nivel del Semillero que va dirigido a estudiantes de tercer semestre y no de tercer año como lo es en la NUS.

II-A. Contenido y estructura de las sesiones

Para el desarrollo del plan de trabajo fue necesario escoger el contenido a trabajar en cada sesión de manera que se trataran de cubrir la mayoría de las técnicas básicas e intermedias de programación requeridas para las maratones. La elección de los temas si hizo basándose en el nivel de los estudiantes del Semillero, el material de los cursos “Competitive Programming” de la NUS [9] y “Escuela de verano para maratones de programación” de la Universidad Estatal de Campinas [6], los temas de varios libros acerca de algoritmos y de competencias de programación [3]–[5], [10], [11] y otros temas que los autores consideraron importantes de acuerdo a su experiencia obtenida gracias a la participación en diferentes competencias de programación.

Cada sesión de Semillero, independiente del tema que se trabaje, consiste en tres partes principales:

1. Discusión, solución y de los problemas propuestos como tarea en la sesión anterior.
Esto se hace con el fin de que los estudiantes, luego de haber intentado resolver los problemas propuestos de manera individual, entiendan su solución y de esta manera aprendan de ella. Para los estudiantes que lograron resolver los problemas este es un espacio en el que pueden compartir su solución con sus compañeros y ver una implementación diferente del problema que resolvieron; esto último les permite conocer diferentes formas de desarrollar y pensar en un mismo algoritmo y posiblemente conocer funciones y métodos del lenguaje C++ que hacen las implementaciones más cortas y sencillas.
2. Exposición del nuevo tema a trabajar, mostrando los algoritmos, los elementos matemáticos relacionados y sus implementaciones en el lenguaje C++ [12].
La forma de abordar los temas para llevar al estudiante al entendimiento del algoritmo está basada principalmente en las métodos de enseñanza manejados en los cursos “Algorithms: Design and Analysis Part 1/2” [13], [14] e “Introduction to Algorithms” [15] y “Competitive Programming” [9]. Por otra parte, la implementación de los algoritmos se basó en las implementaciones mostradas en [3], [4], [11] con el fin de tener implementaciones eficientes y lo más sencillas posibles.
3. Presentación breve de los problemas propuestos como ejercicio para la siguiente sesión.
Los problemas propuestos son seleccionados entre los problemas disponibles en los jueces de programación en línea UVa [16], Codeforces [17] y Spoj [18] buscando que se resuelvan utilizando los temas de la sesión y algunos temas de sesiones anteriores. Dado que el archivo de problemas en estos jueces es bastante extenso, la búsqueda de los problemas según

Semana	Temas
1	Introducción a C++ y a los jueces de programación ¿En qué consiste una maratón de programación? Solución a un problema básico de maratón de programación
2	Arreglos en C++, Vectores de C++ y Grafos PONER TEMA JUAN OJO NO OLVIDAR
3	Representación de grafos en C++ Entrada usando <code>getline</code> y <code>stringstream</code>
4	Pila y Cola Búsqueda en anchura (BFS) Búsqueda en profundidad (DFS)
5	Problemas de BFS y DFS
6	Map, Set, Heap Ordenamiento topológico Componentes fuertemente conexas (SCC)
7	Algoritmo de Dijkstra
8	Algoritmo de Bellman-Ford
9	Programación dinámica: Problemas clásicos
10	Algoritmo de Floyd-Warshall
11	Árbol de mínima expansión
12	Algoritmo de Knuth-Morris-Pratt
13	Algoritmo de máximo flujo
14	Solución de problemas de la IV Maratón de Programación UTP
15	Algoritmos de teoría de números

Tabla I. TEMAS DESARROLLADOS EN EL SEMILLERO

los temas se hace con ayuda de los problemas que proponen Steven y Felix Halim en sus libros “Competitive Programming 1/2” [3], [4], Ahmed Shamsul Arefin en su libro “Art of Programming Constest” [11] y con ayuda de dos buscadores de problemas en los cuales la búsqueda se hace de acuerdo al tema del cual trata el problema que son: el buscador de Codeforces y el un buscador de problemas para el sitio UVa desarrollado por Mark Greve [19]. Luego de seleccionarlos, los problemas deben se resuelven ya que sus soluciones no están disponibles en ninguna de los libros o sitios web mencionados anteriormente.

III. RESULTADOS

III-A. Programa clase a clase

El Semillero se desarrolló en 15 sesiones semanales de dos horas de duración cada una. Los contenidos trabajados en cada sesión se muestran en la tabla I. Aquí se puede ver que la dificultad de los algoritmos va aumentando progresivamente y los elementos necesarios para entender e implementar un algoritmo específico se enseñan antes de dicho algoritmo. Ejemplos de esto serían la necesidad de aprender a utilizar los arreglos y los vectores antes de aprender las formas de representar un grafo, la importancia de conocer y entender el algoritmo de búsqueda en anchura y la estructura de datos del heap antes de implementar el algoritmo de Dijkstra, o explicar cómo funciona la programación dinámica antes de discutir el algoritmo de Floyd-Warshall.

III-B. Diapositivas

Con el fin de que facilitar la comprensión de los temas, de que los estudiantes tuvieran material con el cual pudieran repasar los temas de las secciones de manera independiente y de dejar un legado para las personas que vayan a estar a cargo del Semillero en el futuro, se decidieron crear diapositivas con los contenidos trabajados en cada sesión. El contenido de estas diapositivas se mantiene actualizado en el repositorio público <https://github.com/anaechavarria/SemilleroProgramacion/> y

además se comparte con los estudiantes al final de cada sesión. Durante el transcurso del semestre, se decidió dar a conocer el contenido de este repositorio con el director de las maratones de programación de la Universidad Tecnológica de Pereira (UTP) y el de la Universidad Pontificia Bolivariana (UPB) quienes difundieron la información entre sus estudiantes. Esta información llegó también a manos de estudiantes de la Universidad Sergio Arboleda, quienes escribieron para preguntar si ellos también podían hacer uso del material para prepararse para las maratones de programación. Luego de compartir la información con estas universidades, 8 personas nuevas empezaron a seguir el contenido del repositorio y otras 3 lo marcaron como favorito.

III-C. Competencias

Como se mencionó en la metodología, las competencias son parte fundamental del programa del Semillero. Es por esto que para cada sesión se buscaron y seleccionaron entre 2 y 4 problemas que tuvieran relación con los temas vistos en dicha sesión y se crearon competencias con dichos problemas en los sitios Contests: Factor Común [20] y Virtual Online Contests [21], dos páginas web para crear competencias con problemas de los jueces de programación mostrados en el Semillero. Se crearon un total de 11 competencias, los resultados de cada competencia se muestran en la tabla II. Allí se pueden observar que ocurrieron dos fenómenos importantes, el primero es que la cantidad de personas que participaron en el Semillero se redujo en el transcurso del semestre y el segundo es que en las competencias 6 a 9 el número de problemas resueltos es muy bajo. Estos fenómenos se deben a que a medida que avanza el semestre, los compromisos académicos son mayores lo que hace que se necesite más tiempo para las actividades la Universidad. Como el Semillero es de carácter opcional, muchos estudiantes se ven forzados a dejar de asistir para poder atender los compromisos académicos obligatorios o no tienen suficiente tiempo para resolver los problemas.

III-D. Problemas resueltos

Para el desarrollo del Semillero y de las competencias fue necesario buscar y solucionar un problema de los archivos de los jueces de programación que se ajustaran a los temas discutidos en cada sesión. Se resolvieron un total de 49 problemas; de estos 2 fueron resueltos en reuniones del Semillero, 34 fueron problemas propuestos para las competencias y los 13 restantes fueron los problemas de la IV Maratón de Programación UTP los cuyas soluciones se discutieron en una sesión del Semillero.

III-E. Maratones

Como parte del objetivo del Semillero, se invitó a los estudiantes a participar en el Circuito Colombiano de Maratones de Programación estas son maratones que se realizan a nivel nacional y son preparatorias para la Maratón Nacional de Programación ACIS/REDIS que se realizará en octubre de este año. En dos de las cuatro competencias se tuvo la participación de 2 equipos de estudiantes del Semillero [22].

Por otro lado, el 4 de mayo se llevó a cabo la IV Maratón de Programación UTP que tenía una dificultad básica/intermedia y estaba pensada para el fortalecimiento de los competidores novatos [23]. En esta competencia, un equipo conformado

Competencia	Problemas	Participantes	Problemas resueltos
1	4	agomezl	4
		svanegas	4
		estebanf01	4
		cmejia49	3
		yampyer	3
		SantiSP	2
		jlopera8	1
2	2	srincon2	1
		agomezl	2
		luisponce	2
		svanegas	1
		estebanf01	0
		cmejia49	0
		yampyer	0
3	4	SantiSP	0
		jlopera8	0
		zubieta	4
		svanegas	3
		luisponce	1
		jlopera8	1
		estebanf01	0
4	3	cmejia49	0
		yampyer	0
		zubieta	3
		svanegas	2
5	3	estebanf01	2
		luisponce	2
		svanegas	3
		luisponce	3
6	4	estebanf01	1
		zubieta	1
7	3	svanegas	1
		spalac24	1
8	3	yampyer	0
		estebanf01	0
		svanegas	1
9	2	estebanf01	1
		yampyer	0
10	2	svanegas	0
		estebanf01	0
11	4	svanegas	2
		estebanf01	1
		svanegas	Competencia sin concluir

Tabla II. RESULTADOS DE LAS COMPETENCIAS REALIZADAS

por dos estudiantes del Semillero quedó en cuarto lugar, compitiendo contra equipos de otras universidades de Colombia.

IV. CONCLUSIÓN

V. AGRADECIMIENTOS

- Al Departamento de Ingeniería de Sistemas de la Universidad EAFIT por su aporte financiero para la realización de este proyecto.
- A la Universidad EAFIT por proporcionar el espacio para las reuniones del Semillero.
- Al profesor Francisco Correa por sus observaciones para la elaboración de los reportes.
- Al estudiante Santiago Palacio Gómez por su constante apoyo durante las reuniones.
- A los integrantes del Semillero de Programación por ser nuestra motivación para ser siempre mejores.

REFERENCIAS

- [1] "Asociación Colombiana de Ingenieros de Sistemas (ACIS)." <http://acis.org.co/>.

- [2] “International Collegiate Programming Contest (ICPC).” <http://icpc.baylor.edu/>.
- [3] S. Halim and F. Halim, *Competitive Programming*. Lulu. com, 2010.
- [4] —, *Competitive Programming 2*. Lulu. com, 2011.
- [5] S. S. Skiena, M. A. Revilla, and M. A. Revila, *Programming challenges: The programming contest training manual*. Springer Heidelberg, 2003.
- [6] A. Lopatin, F. Dias Moreira, and F. I. Schaposnik Massolo, “Material de Curso: Escola de Verão - Maratona de Programação 2012 - Instituto de Computação - UNICAMP,” <http://maratona.ic.unicamp.br/MaratonaVerao2012>.
- [7] S. Halim and F. Halim, “Competitive Programming in National University of Singapore,” in *Ediciones Sello Editorial SL (Presented at Collaborative Learning Initiative Symposium (CLIS) ACM ICPC World Final 2010, Harbin, China, 2010*.
- [8] Universidad Nacional de Singapur (NUS), “Algorithmics @ NUS Wiki.” <http://algorithmics.comp.nus.edu.sg/wiki/>.
- [9] S. Halim, “Material de clases del curso Competitive Programming,” <https://sites.google.com/site/stevenhalim/home/material>, <https://sites.google.com/site/stevenhalim/home/material>.
- [10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA, USA: MIT Press, 2009.
- [11] A. S. Arefin, *Art of Programming Contest*. Gyankosh Prokashoni, 2006.
- [12] B. Stroustrup, *El lenguaje de programación C++*. Addison Wesley, 2002.
- [13] T. Roughgarden, “Algorithms: Design and Analysis, Part 1 (Stanford University),” <https://www.coursera.org/course/algo>.
- [14] —, “Algorithms: Design and Analysis, Part 2 (Stanford University),” <https://www.coursera.org/course/algo2>.
- [15] C. Leiserson and E. Demaine, “6.046J Introduction to Algorithms (SMA 5503),” Massachusetts Institute of Technology: MIT OpenCourseWare, <http://ocw.mit.edu>.
- [16] M. A. Rivilla, “UVa Online Judge,” <http://uva.onlinejudge.org>.
- [17] M. Mirzayanoc, “Codeforces,” <http://www.codeforces.com>.
- [18] Sphere Research Labs, “Sphere Online Judge,” <http://www.spoj.com>.
- [19] M. Greve, “UVa Toolkit,” <http://uvatoolkit.com/about.php>.
- [20] A. Mejía Posada, “Contests: Factor Común,” <http://contests.factorcomun.org/>.
- [21] A. Aly, “Virtual Online Contests,” <http://ahmed-aly.com/>.
- [22] “Circuito Colombiano de Maratones de Programación (CCMP),” <http://acm.javeriana.edu.co/maratones/>.
- [23] U. T. de Pereira, “IV Maratón de Programación UTP,” <http://acm.javeriana.edu.co/maratones/2013/05/02/iv-maraton-interna-de-programacion-utp-2013/>.

ANEXOS