Desarrollo e implementación de un programa de trabajo para el semillero de programación

Autor:

Ana Echavarría Uribe

Tutor:

Juan Francisco Cardona Mc'Cormick

Universidad Eafit

Escuela de Ingeniería

fcardona@eafit.edu.co

Universidad Eafit

Ingeniería Matemática

Medellín

2013

Tabla de contenidos

1. Planteamiento del problema **2**

2. Objetivos generales y específicos **2**

2.1. Objetivo general 2

2.2. Objetivos específicos 2

3. Antecedentes **4**

4. Justificación **4**

5. Alcance **5**

6. Metodología propuesta **6**

7. Cronograma de actividades **6**

8. Presupuesto **7**

8.1. Personal 7

8.2. Materiales 7

9. Propiedad intelectual **7**

10. Glosario **8**

1. Planteamiento del problema

El semillero de programación es un grupo de la Universidad que tiene como objetivo presentar nuevas técnicas de programación a estudiantes interesados en mejorar sus habilidades en esta área y prepararlos para participar en las maratones de programación realizadas por ACIS/REDIS [1] y por la ACM-ICPC [2]. En este semillero se discuten los temas más útiles [3, 4, 5, 9] para estas competencias que son 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; sin embargo, durante este tiempo no se desarrolló un plan de trabajo para el semillero. Lo que se busca en esta práctica investigativa es desarrollar e implementar un programa de trabajo para el semillero de programación que le dé a los estudiantes las bases teóricas y prácticas para mejorar sus habilidades y enfrentarse por primera vez a las maratones de programación.

2. Objetivos generales y específicos

2.1. Objetivo general

Desarrollar e implementar un plan de actividades y temas de fundamentación matemática para el semillero de programación que busque mejorar las habilidades de programación de los estudiantes, basándose en la metodología usada por Steven Halim en el curso Competitive Programming de la Universidad Nacional de Singapur y en otras que en el desarrollo del proyecto puedan surgir.

2.2. Objetivos específicos

- Adaptar, al nivel de los estudiantes del semillero de programación, la metodología de clases del curso Competitive Programming dictado por Steven Halim [8] y otras que se puedan ser de utilidad.
- Presentar y documentar los conceptos básicos de la STL (Standard Template Library) de C++.
- Introducir a los estudiantes a algunos jueces en línea como *Codeforces* [10], *UVa* [11] y *Spoj* [12].

- Buscar, resolver y sugerir problemas en los diferentes jueces que permitan a los estudiantes aplicar los conceptos aprendidos en las reuniones y queden como banco de problemas para uso futuro.
- Hacer revisiones de las soluciones a los problemas propuestos luego de que los estudiantes hayan intentado resolverlos de manera independiente, discutiendo los algoritmos y fundamentos matemáticos involucrados.
- Crear material de trabajo (diapositivas, programas, problemas, competencias) con los temas enseñados y compartirlo con los estudiantes para que les sirvan como material de estudio en casa.
- Crear el repositorio *github.com/anaechavarria/SemilleroProgramacion* y actualizarlo cada semana con el material del semillero.
- Presentar y documentar los algoritmos de recorrido de grafos en profundidad (DFS) y en anchura (BFS) y sus aplicaciones.
- Presentar y documentar los algoritmos para hallar los caminos más cortos en grafos: Dijkstra, Bellman-Ford y Floyd-Warshall.
- Presentar y documentar en qué consisten los métodos y técnicas de programación dinámica y los problemas más comunes que se resuelven usándolos.
- Mostrar cómo se pueden hallar las ocurrencias de una palabra en un texto de manera rápida usando el algoritmo de KMP.
- Presentar y documentar los algoritmos de teoría de números más útiles [3, 4, 5, 9] para las competencias de programación.
- Presentar y documentar el algoritmo de Kruskal para hallar el árbol de mínima expansión.
- Presentar y documentar los algoritmos de geometría más usados [3, 4, 5, 9] en las competencias de programación.
- Promover la participación de los estudiantes en las maratones de programación.
- Dejar un legado estructurado y organizado para generaciones futuras con fundamentos matemáticos autocontenidos y estructurados.

3. Antecedentes

El semillero de programación fue establecido hace varios años en la Universidad EAFIT y ha permitido que los estudiantes con interés en la programación y en las matemáticas tengan un espacio para aprender, interactuar y socializar problemas. En los años anteriores, el semillero era un espacio muy productivo para los estudiantes avanzados ya que durante las reuniones se socializaban y resolvían problemas de dificultad media y alta lo que le permitía, a quienes asistían, mejorar su habilidad de solución de problemas. Para los principiantes, sin embargo, el nivel del semillero era avanzado y se trataban problemas que ellos todavía no tenían la fundamentación teórica para resolver.

El semestre 2012-1 se dividió el semillero de programación en dos grupos: básico y avanzado. En el semillero avanzado se siguieron resolviendo y discutiendo problemas como se hacía anteriormente y el semillero básico se enfocó en desarrollar las habilidades de programación de los estudiantes de primero y segundo semestre. Este cambio logró despertar el interés de nuevos alumnos que anteriormente no asistían al semillero porque el nivel era muy avanzado para ellos.

Similar a lo que se ha trabajado en el semillero de programación existe desde 2008 un curso en la Universidad Nacional de Singapur (NUS) llamado Competitive Programming que busca fortalecer las habilidades de programación de sus estudiantes destacados para prepararlos para las competencias universitarias de programación de la ICPC [6]. Este curso está enfocado para estudiantes de tercer año de las áreas de ciencias de la computación, matemáticas e ingeniería electrónica que sean destacados en programación y busca enseñar algoritmos avanzados de manera práctica en lugar de demostrar la corrección de los algoritmos de manera rigurosa.

El curso de la NUS está basado en los libros [4, 5] escritos por Steven Halim donde se explican los algoritmos y sus implementaciones y luego se propone una lista de problemas que se resuelven utilizando esos algoritmos. Por lo anterior, la metodología de este curso consta tanto de clases magistrales como de competencias ya que Steven Halim considera que es parte de la naturaleza humana tratar de lograr los mejores resultados cuando se compite con los demás [6]. Para complementar el proceso, se realizará una búsqueda de experiencias similares y se hará contacto con los autores.

4. Justificación

Actualmente, los estudiantes que pertenecen al semillero de programación son en su mayoría de tercer semestre, teniendo también estudiantes de primer semestre. Lo anterior

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 algoritmos de grafos, strings y geometría, la programación dinámica y los conceptos y algoritmos básicos de teoría de números. Este semestre se busca utilizar el programa de clases que desarrolló Steven Halim y otras metodologías que puedan surgir y adaptarlas para que se ajuste al nivel de los estudiantes del semillero, de manera que tengan la fundamentación teórica necesaria para resolver los problemas más comunes presentados en las maratones de programación.

Por otro lado se busca que la documentación y diapositivas utilizadas este semestre sirvan como material de trabajo para futuras generaciones del semillero. Por otro lado, el plan de trabajo permitirá que aquellos que continúen a cargo del semillero de programación el semestre entrante sepan cuáles son los conceptos que los estudiantes actuales conocen y puedan continuar enseñándoles aquellos temas que los estudiantes todavía no conocen.

El curso de la NUS de Competitive Programming, dictado por Steven Halim, comenzó en agosto de 2008 y muchos de sus estudiantes han logrado resultados favorables en las competencias de la ICPC, obteniendo incluso mención honorífica en Maratón Mundial ACM-ICPC celebrada en Varsovia en 2012 [7]. Al igual que en este curso, 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, como ha ocurrido en ocasiones anteriores.

5. Alcance

La implementación de este proyecto permitirá que los estudiantes participantes en el semillero de programación, en su mayoría pertenecientes a las carreras de Ingeniería de Sistemas e Ingeniería Matemática de la Universidad se motiven y estén preparados para participar en el Circuito de Maratones de Programación ACIS/REDIS y en la Maratón Nacional de Programación ACIS/REDIS. De obtener buenos resultados en esta última competencia, los estudiantes podrían tener la oportunidad participar bajo el nombre de la Universidad en la Maratón Regional Suramericana ACM-ICPC y en la Maratón Mundial ACM-ICPC.

6. Metodología propuesta

El semillero de programación consiste en una reunión semanal que dura entre 1 hora y media y 2 horas. La estructura de cada reunión se compone de tres etapas principales:

1. Discusión, solución y revisión de los problemas propuestos como tarea en la sesión anterior.
2. Exposición del nuevo tema a trabajar, mostrando los algoritmos y elementos matemáticos relacionados.
3. Presentación breve de los problemas propuestos como ejercicio para la siguiente sesión.

Para desarrollar cada reunión es necesario investigar acerca del tema a enseñar, preparar documentación (diapositivas, programas, competencias) que le permita a los estudiantes repasar lo aprendido y buscar y solucionar problemas que se ajusten al nivel y los temas aprendidos en la clase.

7. Cronograma de actividades

El semillero consistirá de 15 reuniones que se celebrarán todos los viernes desde el 1 de febrero hasta el 17 de mayo de 2013. Los temas a tratar en cada reunión son:

Fecha	**Tema**
Febrero 1	Introducción a C++ y al juez de programación Codeforces [10] ¿En qué consiste una maratón de programación? Solución a un problema básico de maratón de programación
Febrero 8	Arreglos en C++, Vectores de C++ y Grafos
Febrero 15	STL de C++
Febrero 22	Algoritmos de recorrido de grafos: BFS y DFS
Marzo 1	Aplicaciones de los algoritmos de BFS y DFS
Marzo 8	Ordenamiento Topológico y Componentes Fuertemente Conexas
Marzo 15	Algoritmos de Dijkstra y Bellman-Ford
Marzo 22	¿Qué es la programación dinámica? Soluciones básicas con programación dinámica Algoritmo de Floyd-Warshall

Abril 5	Aplicaciones de la programación dinámica: Longest Common Substring (LCS) y 0-1 Knapsack
Abril 12	Algoritmo de Knuth-Morris-Pratt
Abril 19	Algoritmos básicos de teoría de números
Abril 26	Solución de problemas de repaso
Mayo 3	Algoritmo de Kruskal para hallar el árbol de mínima expansión (MST)
Mayo 10	Algoritmo de Edmonds-Karp para hallar el máximo flujo
Mayo 17	Algoritmos de geometría

8. Presupuesto

8.1. Personal

La dedicación de la estudiante Ana Echavarría Uribe al proyecto será de 10 horas a la semana entre el 28 de enero y el 25 de mayo de 2013. Estas horas serán remuneradas al valor que tenga establecido la Universidad EAFIT para las monitorías y el costo será asumido por el Departamento de Ingeniería de Sistemas.

8.2. Materiales

Para la realización del proyecto será necesario el uso de un aula de clases cada semana durante dos horas. Esta aula será proporcionada por la Universidad EAFIT.

9. Propiedad intelectual

Debido a que este trabajo es una actividad continua de la Universidad, el producto de este trabajo pertenece le en un 100 %. El contenido del trabajo podrá ser utilizado en su totalidad para el desarrollo del semillero de programación de la Universidad sin previo aviso a los autores.

10. Glosario

ACIS/REDIS Alianza entre la Asociación Colombiana de Ingenieros de Sistemas (ACIS) y la Red de Decanos y Directores de Ingeniería de Sistemas y Afines (REDIS) que organiza actividades alrededor de Maratones de Programación en Colombia y Suramérica [1].

ACM-ICPC (International Collegiate Programming Contest) es la entidad organizadora de los concursos internacionales de programación para universitarios [2].

STL (Standard Template Library) es una librería de C++ que provee contenedores, funciones y algoritmos que pueden ser utilizadas con cualquier tipo de dato [14].

Juez en línea Sistema en línea que hace un juzgamiento o evaluación automática de un conjunto de problemas de programación que se encuentren en su base de datos.

Revisión Discusión de un código por parte de un grupo de programadores con el fin de aprender las técnicas que se implementan allí.

Repositorio Estructura de datos, almacenada en un servidor, que contiene, entre otras cosas, un conjunto de archivos y el historial de los cambios hechos a estos archivos.

BFS (Breath-First Search) es un algoritmo de búsqueda de grafos que hace un recorrido en anchura, es decir, visita todos los nodos a distancia k del nodo de inicio antes de visitar cualquier nodo a distancia $k + 1$ [13].

DFS (Depth-First Search) es un algoritmo de búsqueda de grafos que hace un recorrido en profundidad, es decir, se visitan todos los nodos salientes del nodo visto más recientemente antes de visitarlo a él [13].

KMP Algoritmo desarrollado por Knuth, Morris y Pratt que hace una búsqueda de una palabra en un texto en tiempo lineal [13].

Bibliografía

[1] Asociación Colombiana de Ingenieros de Sistemas (ACIS). http://acis.org.co/index.php?id=556

[2] International Collegiate Programming Contest (ICPC). http://icpc.baylor.edu/

[3] SKIENA, S.S., REVILLA, M.A. *Programming Challenges: The Programming Contest Training Manuals.* Springer, 2003.

[4] HALIM, S., HALIM, F. *Competitive Programming: Handbook for ACM ICPC and IOI Contestants*. Lulu, 2010.

[5] HALIM, S., HALIM, F. *Competitive Programming 2: Handbook for ACM ICPC and IOI Contestants*. Lulu, 2011.

[6] HALIM, S., HALIM, F. *Competitive Programming in National University of Singapore*. Department of Computer Science, School of Computing, National University of Singapore.

[7] Universidad Nacional de Singapur (NUS) *Algorithmics @ NUS Wiki*. http://algorithmics.comp.nus.edu.sg/wiki/

[8] HALIM, S. Material de clases del curso Competitive Programming. https://sites.google.com/site/stevenhalim/home/material

[9] LOPATIM, A., DIAS MOREIRA, F., SCHAPOSNIK MASSOLO, F. I. *Material de Curso: Escola de Verão - Maratona de Programação 2012*. Instituto de Computação - UNICAMP. http://maratona.ic.unicamp.br/MaratonaVerao2012/

[10] MIRZAYANOV, M. *Codefroces*. http://www.codeforces.com

[11] RIVILLA, M.A. *UVa Online Judge*. http://uva.onlinejudge.org/

[12] Sphere Research Labs. *Sphere Online Judge*. http://www.spoj.com/

[13] CORMEN, T.H., LEISERSON, C.E., RIVEST, R.L., STEIN, C. *Introduction to Algorithms*. The MIT Press, 3ra ed. 2009.

[14] STROUSTRUP, B. *El Lenguaje de Programación C++*. Addison Wesley. 2002.