CC50R Taller de Implementación de Estructuras de Datos y Algoritmos
(10 UD)

Semestre 2005/2
Prof. Rodrigo Paredes

Requisitos: CC40A / Autor

- **Objetivos:**

El objetivo del curso es que los alumnos apliquen los conocimientos aprendidos en los cursos de Algoritmos (a saber cc30a y cc40a). Para esto, el énfasis del curso está en la resolución de una gran variedad de problemas usando las técnicas estándares de diseño de soluciones algorítmicas, tanto en forma personal como en grupos pequeños. Al finalizar el curso, se espera que los alumnos ganen experiencia práctica tanto en el diseño como en la implementación de algoritmos y estructuras de datos de dificultad media. Adicionalmente, se espera preparar y escoger equipos para participar en el concurso de programación ACM-ICPC.

- **Metodología:**

Durante el primer mes de clases se revisarán las técnicas estándares de diseño de algoritmos, y luego algunas estructuras de datos y algoritmos de uso frecuente, haciendo énfasis en los aspectos de implemetación.

A continuación se revisarán una gran variedad de problemas específicos (obtenidos del torneo ACM-ICPC), de modo de poder aplicar en contexto las técnicas revisadas anteriormente.

Interesan implementaciones que sean eficientes y robustas (con respecto a que resistan los casos de borde del problema). Las clases se realizarán en sala con computador.

- **Contenidos:**

- Técnicas básicas:
  Backtracking, programación dinámica, algoritmos greedy, recursión, inducción, divide & conquer y algoritmos en línea.

- Herramientas generales:
  Algoritmos de ordenamiento en memoria principal y secundaria; implementaciones de conjuntos: arreglos extensibles, tablas de hashing, hashing extensible; diccionarios: árboles binarios, árboles digitales, árboles 2-3, arreglos asociativos; colas de prioridad: treaps, heaps de acceso aleatorio, min-max-heaps; grafos: representación matricial, lista de adyacencia, caminos mínimos y árbol cobertor mínimo.

- Combinaciones de las técnicas anteriores.

* **Programa :**

  * Introducción: visión general del curso, revisión algunos problemas representativos (1.5 horas).

  * Técnicas básicas de diseño de algoritmos (3 horas).

  * Revisión de las herramientas generales (10.5 horas).

  * Revisión de los problemas de la competencia ACM-ICPC (24 horas):
    Expresiones regulares, representaciones numéricas, ordenamiento, selección, problemas combinatoriales, problemas de optimización, problemas geométricos y problemas de grafos.

* **Evaluación**

  La evaluación consiste en varias tareas individuales y en pequeños grupos. Las herramientas generales se evaluarán en forma individual, los problemas de la competencia en forma grupal. Algunas de las tareas (en ambas etapas del curso) serán revisadas en presencia del (los) alumno(s), de modo de revisar los detalles de implementación.

  Las evaluaciones individuales y grupales deben ser aprobadas independientemente. La evaluación individual consiste en la realización de 3 proyectos pequeños que combinen al menos 2 de las herramientas generales. Se espera que resuelvan al menos 20 problemas de la competencia ACM-ICPC. Las ponderaciones son 1/3 para la evaluación individual y 2/3 para la grupal.

  OBS: Los mejores grupos serán seleccionados para participar en la competencia ACM-ICPC.

- **Bibliografía**

* The ACM-ICPC International Collegiate Programming Contest Web Site sponsored by IBM, http://icpc.baylor.edu/icpc/

* The 2000's ACM-ICPC Live Archive Around the World, http://acmicpc-live-archive.uva.es/nuevoportal/

* Teamwork in Programming Contests: 3 * 1 = 4, http://www.acm.org/crossroads/xrds3-2/progcon.html

* C. Collazos, A Methodology for the Computational Support of Evaluation and Monitoring in Collaborative Learning Environments. Doctoral dissertation, Department of Computer Science, Universidad de Chile. Santiago, Chile, 2003.

* T. Cormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to Algo-rithms. The MIT Press, 2 edition, 2001.

* Bruce Eckel. Thinking in C++, Volume 1: Introduction to Standard C++ (2nd Edition). Prentice Hall.

* Eckel, Bruce. Thinking in Java (3rd Edition). Prentice Hall PTR; 3 edition.