Instituto Tecnológico de Costa Rica

Escuela de Computación

IC 3002 Análisis de Algoritmos

Segundo semestre de 2016

Proyecto válido por el primer examen parcial

Descripción general del problema: dado un número entero impar menor que una cierta cota, decidir si es primo o no.

Trabajo de los estudiantes: escribir diferentes programas que solucionen este problema desde diferentes perspectivas de diseño de algoritmos.

Importancia del problema: los números primos tienen una gran importancia para el diseño de llaves asimétricas en algunos algoritmos criptográficos, así como para funciones hash criptográficas. Además, su enorme importancia en Álgebra hace que sean muy útiles en el diseño de códigos de corrección de errores (v. gr. Reed-Solomon) o de funciones hash comunes. En muchas de estas aplicaciones se requieren números enteros que no solo sean primos sino que tengan algunas otras propiedades, por lo que "primalidad" suele ser apenas el primer paso en estas aplicaciones. Por otro lado, esta prueba para los estudiantes introduce el uso de los mapas de bits como una estructura de datos práctica para la "memoización" que requieren los algoritmos de programación dinámica.

Descripción precisa del trabajo de los estudiantes: deberán hacer tanto trabajo teórico como práctico. El trabajo práctico consiste en crear dos programas, uno sencillo y otro no tan sencillo. Ambos programas deberán recibir interactivamente del usuario un número entero, si es par se dicta el veredicto inmediatamente, si es impar se deberá verificar si el número es primo o no. Una vez hecho esto, el programa espera un nuevo número o la indicación de que debe terminar, ambas cosas mediante una interfase gráfica de usuario fácil de usar. Los programas deberán ser capaces de decidir si un número entero impar de hasta 12 dígitos decimales.

1. Programas a diseñar
    a. Primer programa: mediante simple división por todos los impares previos, eso sí, no más allá de lo estrictamente necesario, el programa dará su veredicto.
    b. Segundo programa: este segundo programa necesita un parámetro adicional, y es el tamaño del entero a comprobar. Debido a que se espera que el programa trabaje sobre una lista de números dados por el usuario, el parámetro que se pide al usuario involucra al mayor número esperado de la serie. El tamaño del espacio a emplear (de un cierto máximo) se calculará con la misma información teórica usada para el Primer Programa. Entonces, este programa tendrá una fase de preparación previa mientras prepara la estructura de datos, y luego procesará uno por uno los números dados por el usuario.

2. Problemas teóricos a resolver
    a. Para ambos problemas se debe estimar el O() del algoritmo empleado. En el caso del segundo programa de deberá estimar por aparte el O() de la construcción del bit map y de la ejecución del test sobre el número dado.
    b. Utilizando datos empíricos y los análisis de complejidad asintótica, determinar cuál programa sería mejor para la tarea.

Restricciones del proyecto:

a. Se podrá efectuar en equipos de hasta dos personas.
b. Se hará una revisión previa el Jueves 29 de Septiembre. El horario de revisiones será de 11:00 a.m. a 5:00 p.m. Si el proyecto no pasa por la revisión previa no podrá ser entregado de manera definitiva.
c. La entrega definitiva será el Martes 2 de Agosto. En ella los estudiantes deberán defender su proyecto personalmente. La defensa del proyecto se hará de 11:30 a.m. a 5:00 pm
d. La evaluación se dividirá de la siguiente forma:
    a. Demostración de que los programas escogen siempre de manera óptima el número de divisiones a efectuar (en un caso) o el tamaño del bitmap, en otro. Si se utilizara alguna propiedad de los enteros deberá demostrarse esta formalmente. (15%)
    b. Funcionamiento correcto del programa 1 (35%)
    c. Funcionamiento correcto del programa 2 (40%)
    d. Demostración correcta de la complejidad asintótica solicitada (10%)
e. Para las citas de revisión se pondrá una lista en la oficina del profesor para que los estudiantes selecciones el “slot” que más les parezca conveniente.