

ALGORITMOS Y ESTRUCTURAS DE DATOS II

GRADO INGENIERÍA INFORMÁTICA, CURSO 14/15

PRÁCTICA DE DIVIDE Y VENCERÁS Y ANÁLISIS DE ALGORITMOS

Subgrupo 1.3

A. Contexto

Las **técnicas generales de diseño de algoritmos** ofrecen un amplio y variado conjunto de herramientas que se pueden usar en la resolución de muchos tipos de problemas. La aplicación de una técnica concreta debe entenderse como un proceso metódico, que empieza con la interpretación y el modelado del problema desde la perspectiva de esa técnica, sigue con la definición de las partes genéricas del esquema algorítmico (tipos de datos y funciones básicas), y acaba con la implementación, prueba, refinamiento y optimización del algoritmo.

En esta práctica se trata de aplicar la técnica divide y vencerás en la resolución de problemas de recorrido de cadenas.

B. Enunciado de la actividad

Esta actividad se hará en grupos de dos alumnos (en casos excepcionales en grupos de un alumno, para lo que habrá que pedir permiso al profesor). Cada grupo tiene asignado un problema distinto. Cada grupo mandará un correo al profesor de prácticas (nmarin@um.es) indicando para cada alumno el nombre, grupo, subgrupo y correo electrónico, y el profesor les contestará asignándoles uno de los problemas con los que trabajar, de entre los enumerados al final de este enunciado.

Se trata de resolver y estudiar el problema en el caso en que se utiliza un divide y vencerás con división del problema en dos subproblemas. La nota máxima en la práctica realizándola con división en dos subproblemas es un OCHO. Opcionalmente se puede generalizar la resolución y estudio realizando la práctica con división del problema en s subproblemas, con s un parámetro del algoritmo. La práctica constará de:

1. Diseñar una solución utilizando la técnica divide y vencerás para resolver el problema. En el caso básico se hará dividiendo el problema en dos subproblemas, y en el caso general con división en s subproblemas, con s un parámetro del algoritmo.
2. Analizar de forma teórica el tiempo de ejecución, en los casos mejor, peor y promedio, del algoritmo obtenido y el orden de ejecución.
3. Diseñar un proceso de validación del algoritmo implementado. Hay que indicar claramente los experimentos realizados para asegurarnos de que el programa funciona correctamente.
4. Implementar un generador de casos de prueba, que genere entradas para el programa para los casos más favorable, más desfavorable y promedio, y realizar un estudio experimental de los tiempos de ejecución.
5. Contrastar los resultados teóricos con los experimentales.

Es requisito indispensable programar el algoritmo propuesto en C/C++ y que funcione correctamente con los casos de prueba que pase el profesor en la revisión de la práctica. No obstante, el funcionamiento correcto (en el sentido de encontrar la solución que se pide) no es condición suficiente para aprobar la actividad, sino que habrá que tener en cuenta otras cuestiones, como eficiencia, análisis teórico y experimental, documentación...

C. Memoria de la actividad

La memoria entregada deberá contener en la portada el nombre de los alumnos, grupo, subgrupo y e-mail.

Para el problema asignado al grupo, se deberán incluir los siguientes apartados (**hay que obtener al menos un tercio de la nota en cada apartado**):

1. (hasta **1.5** puntos el básico y **2** el generalizado) Pseudocódigo y explicación del algoritmo, justificando las decisiones de diseño, la estructuras de datos y las funciones básicas del esquema algorítmico.
2. (hasta **1.5** puntos el básico y **2** el generalizado) Estudio teórico del tiempo de ejecución del algoritmo (t_m , t_M y t_p) y de la ocupación de memoria (m_m , m_M y m_p) y obtención de conclusiones acerca de los órdenes.
3. (hasta **1.5** puntos el básico y **2** el generalizado) Programación del algoritmo. El programa debe ir documentado, con explicación de qué es cada variable, qué realiza cada función y su correspondencia con las funciones básicas del esquema algorítmico correspondiente.
4. (hasta **1** punto) Validación del algoritmo, justificando los experimentos realizados para asegurar que el algoritmo funciona correctamente, y en su caso programas utilizados para la validación.
5. (hasta **1.5** puntos el básico y **2** el generalizado) Estudio experimental del tiempo de ejecución para distintos tamaños de problema. Habrá que experimentar con tamaños suficientemente grandes para obtener resultados significativos. Se debe analizar la influencia del tamaño del caso base en el tiempo de ejecución. En el caso generalizado habrá que estudiar cómo influye el valor del parámetro s en el tiempo de ejecución.
6. (hasta **1** punto) Contraste del estudio teórico y el experimental, buscando justificación a las discrepancias entre los dos estudios.

En un apartado final se incluirán las conclusiones y valoraciones personales de la actividad, y una estimación del tiempo total que se ha tardado en completarla, distinguiendo entre tiempo dedicado a la versión básica y a la generalización.

D. Evaluación de la actividad

La documentación generada se enviará por correo en pdf, cada grupo a su profesor de prácticas, a su correo personal o a través del aula virtual. La

fecha tope de entrega de la documentación será el 13 de abril de 2015, durante todo el día.

Los profesores realizarán entrevista individual con cada uno de los alumnos, aunque se puede decidir no realizar la entrevista con alguno de ellos si ha seguido su trabajo en las sesiones de prácticas. La fecha de la entrevista se fijará tras el envío de la documentación.

Copiar cualquier ejercicio de otro grupo supondrá el suspenso fulminante, no sólo de la práctica sino de **toda la asignatura**, para todos los alumnos implicados.

E. Problemas

ALFABETO para todos los problemas: abcdefghijklmnopqrstuvwxyz

- 1) Dada una cadena C con n caracteres y un conjunto S de 5 caracteres, se trata de encontrar las subcadenas de C formadas por 3 elementos de S. Habrá que obtener el número de subcadenas y su posición en la cadena C. No se considerarán subcadenas incluidas en otras más largas. Por ejemplo, si

C = abbfabcddfcbbade , n=16

si consideramos un conjunto de cinco caracteres $S=\{a,b,c,d,e\}$

la solución es 2, en las posiciones abbfabcddfcbbade

la última cadena, ade, no se tiene en cuenta, pues no se consideran repeticiones de caracteres en las cadenas, y de las cadenas que solapan elementos se considera únicamente la primera en la cadena C.

- 2) Dada una cadena C con n caracteres y un conjunto S de 5 caracteres, se trata de encontrar las subcadenas de C formadas por los 5 elementos de S sin repetir. Habrá que obtener el número de subcadenas y su posición en la cadena C. No se considerarán subcadenas incluidas en otras más largas. Por ejemplo, si

C = acbfabcedfcbcbadec , n=18

si consideramos un conjunto de cinco caracteres $S=\{a,b,c,d,e\}$

la solución es 2, en las posiciones acbfabcedfcbcbadec

la última cadena, badec, no se tiene en cuenta, pues no se consideran repeticiones de caracteres en las cadenas, y de las cadenas que solapan elementos se considera únicamente la primera en la cadena C.

- 3) Dada una cadena C con n caracteres, se trata de encontrar las subcadenas de entre 3 y 8 caracteres de C formadas por caracteres en orden alfabético creciente (consideramos también caracteres repetidos). Habrá que obtener el número de subcadenas y su posición en la cadena

C. No se considerarán subcadenas incluidas en otras más largas. Por ejemplo, si

C = abbcabddacbbad , n=15

la solución es 2, en las posiciones abbc abcdd acbbad

- 4) Dada una cadena C con n caracteres, se trata de encontrar las subcadenas de 5 caracteres de C formadas por caracteres en orden alfabético tanto creciente como decreciente (consideramos también caracteres repetidos). Habrá que obtener el número de subcadenas y su posición en la cadena C. No se considerarán subcadenas incluidas en otras más largas. Por ejemplo, si

C = abbcabddacddccbb , n=16

la solución es 2, en las posiciones abbc abcdd ac dccbb

- 5) Dada una cadena C con n caracteres y 3 subcadenas de longitud 3, se trata de encontrar las apariciones de concatenaciones de dos de las subcadenas en la cadena C. Habrá que obtener el número de subcadenas de C y su posición en la cadena C. No se tendrán en cuenta superposiciones de cadenas, y en caso de superposición se considerarán las concatenaciones más al principio de la cadena C. Por ejemplo, si

C = abbcabddacbdcaaac , n=18

subcadenas acb , aac , dca

la solución es 1, en la posición abbcabddacbdcaaac

- 6) Dada una cadena C con n caracteres, se trata de encontrar la subcadena más larga compuesta por consonantes seguidas de vocales. Por ejemplo, si

C = abccabeaddededebdae

el resultado es abccabeaddededeebdae

- 7) Dada una cadena C con n caracteres, encontrar la subcadena S más larga compuesta por concatenación de subcadenas de la forma $c_1c_2c_3v_1v_2$, donde c_i ($i=1,2,3$) es una consonante y v_j ($j=1,2$) una vocal.

Indicar el comienzo de la cadena y el número de caracteres. En caso de empate devolver la que esté más a la izquierda.

Por ejemplo, si C = abccaebcdeadddaedebddae

el resultado es abccaebcdeadddaedebddae

8) Dada una cadena C con n caracteres, encontrar la subcadena S más larga de C tal que cualquier subcadena de 3 elementos $e_1e_2e_3$ de S cumpla que la suma de los valores de los tres elementos esté en el rango [A, B], con $A < B$. Se entiende por valor de un carácter a su posición en el alfabeto, $\text{valor}(a)=1$, $\text{valor}(b)=2$, etc.

Indicar el comienzo de la cadena y el número de caracteres. En caso de empate devolver la que esté más a la izquierda.

Por ejemplo, si $A=3$, $B=6$ y $C = \text{xabcafxyzabcdstr}$

el resultado es xabcafxyzabcdstr

9) Dada una cadena C con n caracteres, encontrar la subcadena S más larga que sea un palíndromo, es decir, tal que la cadena se lea igual de izquierda a derecha que de derecha a izquierda.

Indicar el comienzo de la cadena y el número de caracteres. En caso de empate devolver la que esté más a la izquierda.

Por ejemplo, si $C = \text{abcabcxyxcbaabc}$

el resultado es abcabcxyxcbaabc

10) Dada una cadena C con n caracteres, encontrar la subcadena S más larga tal que la diferencia entre el valor de dos caracteres consecutivos de S no sea mayor que 3. Se entiende por valor de un carácter a su posición en el alfabeto, $\text{valor}(a)=1$, $\text{valor}(b)=2$, etc.

Indicar el comienzo de la cadena y el número de caracteres. En caso de empate devolver la que esté más a la izquierda.

Por ejemplo, si $C = \text{abcehfeksrtzyx}$

el resultado es abcehfeksrtzyx

11) Dada una cadena C con n caracteres, encontrar el número de subcadenas no descendentes (es decir, tales que nunca hay un carácter seguido de otro que sería alfabéticamente anterior) de al menos dos caracteres, sin tener en cuenta aquellas subcadenas que están contenidas dentro de otra no descendente más larga.

Devolver, además del número, cuáles son las subcadenas, indicando para cada una su comienzo y el número de caracteres.

Por ejemplo, si $C = \text{abcehfeksrtzyxdbcehaeksrtzyx}$

el resultado es 6 (abceh, eks, rtz, bceh, aeks, rtz).

12) Dada una cadena C con n caracteres, y un vector de enteros D de m elementos, encontrar el número de subcadenas de tamaño $m+1$ tales que las diferencias entre los valores de sus elementos consecutivos coinciden

con los valores de D. Se entiende por valor de un carácter a su posición en el alfabeto, valor(a)=1, valor(b)=2, etc.

Devolver, además del número, cuáles son las subcadenas, indicando para cada una su comienzo y el número de caracteres.

Por ejemplo, si C = xabefijmnpqxyzabc y D=[1, 3, 1, 3]

el resultado es 3 (xabefijmnpqxyzabc, xabefijmnpqxyzabc y xabefijmnpqxyzabc).