

# ALGORITMOS Y ESTRUCTURAS DE DATOS II

## GRADO INGENIERÍA INFORMÁTICA, CURSO 16/17

### PRÁCTICA DE DIVIDE Y VENCERÁS Y ANÁLISIS DE ALGORITMOS

#### 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 se permitirán grupos de un alumno, para lo que habrá que pedir permiso al profesor). En cualquier caso, el trabajo a realizar será el mismo.

Cada grupo tendrá asignado un problema distinto. El profesor de prácticas correspondiente será quien asigne a cada grupo uno de los problemas con los que trabajar de entre los enumerados al final de este enunciado.

La práctica consistirá en resolver y estudiar el problema asignado en el caso en que se utiliza un divide y vencerás con división del problema en dos subproblemas. La nota máxima de la práctica básica es un 8, pudiéndose optar hasta a un 11 mediante las opciones que se indican más adelante.

#### La práctica básica constará de los siguientes puntos:

1. **Diseño** de una solución utilizando la técnica divide y vencerás para resolver el problema, dividiendo el problema en dos subproblemas.
2. Análisis teórico del tiempo de ejecución, en los casos mejor, peor y promedio, y del orden de ejecución del algoritmo obtenido.
3. Diseño de un proceso de validación del algoritmo implementado. Hay que indicar claramente los experimentos concretos realizados para asegurarnos de que el programa funciona correctamente.
4. Implementación de un generador de casos de prueba, que genere entradas para el programa para los casos más favorable, más desfavorable y promedio, y realización de un estudio experimental de los tiempos de ejecución.
5. Contraste de los resultados teóricos con los experimentales.

El grupo que quiera optar a una mayor nota podrá resolver además una o dos de las opciones siguientes:

- Estudio y resolución del problema asignado dividiéndolo en tres subproblemas (además de en dos). Consistiría en trabajar los cinco puntos de la práctica con las nuevas condiciones. Valoración: hasta 2 puntos.

- Estudio del problema original generando las entradas con un nuevo alfabeto más reducido de 6 caracteres diferentes (además del alfabeto completo). Esto implicaría trabajar los puntos 2, 4 y 5 de la práctica. Valoración: hasta 1 punto.

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) Pseudocódigo y explicación del algoritmo, justificando las decisiones de diseño, las estructuras de datos y las funciones básicas del esquema algorítmico.
2. (hasta **1,5** puntos) Estudio teórico del tiempo de ejecución del algoritmo ( $t_m$ ,  $t_M$  y  $t_p$ ) y obtención de conclusiones acerca de los órdenes.
3. (hasta **1,5** puntos) 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) Estudio experimental del tiempo de ejecución para distintos tamaños de problema. Habrá que experimentar con tamaños suficientemente grandes para obtener resultados significativos. En el caso de resolver la opción a) habrá que estudiar cómo influye en el tiempo de ejecución hacer 2 o 3 subdivisiones.
6. (hasta **1** punto) Contraste del estudio teórico y el experimental, buscando justificación a las discrepancias entre los dos estudios.

Si se resuelve alguna de las dos opciones adicionales, la puntuación extra se distribuirá entre estos apartados de forma proporcional a su puntuación original.

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 los apartados opcionales.

## 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 (según se establezca). La fecha tope de entrega de la documentación será el 3 de abril de 2017, durante todo el día.

Los profesores realizarán entrevista con cada uno de los grupos, 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 o de otra fuente supondrá el suspenso fulminante, no solo de la práctica sino de **toda la asignatura**, para todos los alumnos implicados.

## E. Problemas

**ALFABETO para todos los problemas: abcdefghijklmnopqrstuvwxyz**

1) Dadas dos cadenas A y B de la misma longitud n, se trata de encontrar la zona de m caracteres consecutivos en las dos cadenas con más caracteres coincidentes y dar el número de caracteres en que coinciden. Cuando haya más de una solución, será válido devolver cualquiera de ellas.

Ejemplo: n=10, m=5

A= c d d a b c d a c c

B= c a c d d b c a d c

Solución: subcadenas que empiezan en la posición 6, y número de coincidencias 2.

---

2) Dadas dos cadenas A y B de la misma longitud n, se trata de encontrar la zona de m caracteres consecutivos en las dos cadenas con más diferencia total en valor absoluto entre los caracteres en cada posición, y dar la mayor diferencia total. Cuando haya más de una solución, será válido devolver cualquiera de ellas.

Ejemplo: n=10, m=5

A= c d d a b c d a c c

B= c a c d d b c a d c

dife: 0 3 1 3 2 1 1 0 1 0

Solución: subcadenas que empiezan en la posición 2, y diferencia total 10.

---

**3)** Dada una cadena A de longitud n, se trata de encontrar la subcadena de m caracteres consecutivos con más caracteres consecutivos ordenados ascendentemente (incluyendo caracteres iguales), y dar el número de caracteres en la mayor subcadena ascendente. Cuando haya más de una solución, será válido devolver cualquiera de ellas.

Ejemplo: n=10, m=5

A= c d d a b c d a c c

Solución: subcadena que empieza en la posición 4, y número de caracteres en la mayor ascendente 4 (a b c d).

---

**4)** Dada una cadena A de longitud n, se trata de encontrar la subcadena de m caracteres consecutivos con más diferencia total en valor absoluto entre caracteres consecutivos, y dar el valor de la diferencia total máxima. Cuando haya más de una solución, será válido devolver cualquiera de ellas.

Ejemplo: n=10, m=5

A= c d d a b c d a c c

Diferencias entre caracteres consecutivos: 1 0 3 1 1 1 3 2 0

Solución: subcadena que empieza en la posición 5, y diferencia total máxima 7.

---

**5)** Dada una cadena A de longitud n y un carácter C, se trata de encontrar la subcadena de m caracteres consecutivos con más apariciones del carácter C y dar el número de veces que aparece el carácter en la subcadena. Cuando haya más de una solución, será válido devolver cualquiera de ellas.

Ejemplo: n=10, m=5, C=c

A= c d d a b c d a c c

Solución: subcadena que empieza en la posición 6, y número de apariciones 3.

---

**6)** Dada una cadena A de longitud n y un carácter C, se trata de encontrar la subcadena de m caracteres consecutivos con más apariciones consecutivas del carácter C y dar el número de veces que aparece el carácter consecutivamente en la subcadena. Cuando haya más de una solución, será válido devolver cualquiera de ellas.

Ejemplo: n=10, m=5, C=c

A= c d d a b c d a c c

Solución: subcadena que empieza en la posición 6, y número de apariciones consecutivas 2.

---

**7)** 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. Por ejemplo, si

C = abbfabccddfcbbade , 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.

---

**8)** 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. 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.

---

**9)** 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 ni total ni parcialmente. Por ejemplo, si

C = abbcabcddacbbad , n=15

la solución es 2, en las posiciones abbc abcdd acbbad

---

**10)** 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 = abbcabcddacdccb , n=16

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

---

**11)** 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 = abbcabcddacbdcaaac , n=18

subcadenas acb , aac , dca

la solución es 1, en la posición abbcabcddacbdcaaac

---

**12)** 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 abccabeaddededebdae

Cuando haya más de una solución, será válido devolver cualquiera de ellas.

---

**13)** 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 = abccaebcdeaddddaedebddae

el resultado es abccaebcdeaddddaedebddae

---

**14)** 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, valor(a)=1, 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 = abcehfeksrtzyx

el resultado es abcehfeksrtzyx

---

**15)** Dada una cadena C con n caracteres, se trata de encontrar la subcadena de m caracteres consecutivos que contenga más caracteres iguales.

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

Por ejemplo, si C = abceeeeksrrzzzyx, m=5

el resultado es 4 caracteres iguales y comienzo en 3 (abceeeeksrzzzyx)

---

**16)** Dada una cadena C con n caracteres, se trata de encontrar la subcadena con m caracteres consecutivos en la que el número de caracteres consecutivos que se diferencia en 2 sea máximo.

Indicar el comienzo de la cadena y el número de caracteres consecutivos que no se diferencian en más de 2. En caso de empate devolver la que esté más a la izquierda.

Por ejemplo, si C = abcehfeksrtzyxdbcehaeksrtzyx, m=5

el resultado comienzo de la cadena 1 y número de caracteres 2 (abcehfeksrtzyxdbcehaeksrtzyx).

---

**17)** Dada una cadena C con n caracteres, y dos caracteres distintos c1 y c2, se trata de encontrar la subcadena de m caracteres consecutivos que contenga más veces el par c1c2.

Indicar el comienzo de la cadena y el número de veces que aparece el par c1c2. En caso de empate devolver la que esté más a la izquierda.

Por ejemplo, si C = abceabcabekssabzzzyx, m=6, c1=a, c2=b

el resultado es comienzo en 1 y número de veces que aparece el par 2 (abceabcabekssabzzzyx)

---

**18)** Dada una cadena C con n caracteres, y dos conjuntos de caracteres C1 y C2, se trata de encontrar la subcadena con m caracteres consecutivos con mayor número de pares formados por un carácter de C1 seguido por otro de C2.

Indicar el comienzo de la cadena y el número de pares del tipo mencionado. En caso de empate devolver la que esté más a la izquierda

Por ejemplo, si C = abceabcabekssabzzzyx, m=6, C1={a,b}, C2={c,d}

el resultado comienzo de la cadena 2 y número de pares 2 (abceabcabekssabzzzyx).