Tarea 2 de Programación 3
11 de octubre de 2024

Se debe entregar un archivo con formato PDF con las soluciones de los ejercicios propuestos. El nombre del archivo debe ser **Tarea2_<Grupo>.pdf**, donde <Grupo> debe ser sustituido por el número que le corresponde al grupo.

Cada grupo debe realizar la entrega en el receptor correspondiente al horario de monitoreo que le fue asignado. Los receptores se encuentran en la pestaña Evaluaciones del sitio del curso.

El plazo para la entrega es el **lunes 21** de octubre a las **8:00 AM**.

Para resolver cada parte puede usar los enunciados de las anteriores y resultados estudiados durante el curso, salvo que se indique lo contrario.

La tarea entregada debe ser creación propia del grupo y estará sujeta al Reglamento de No Individualidad en instancias de evaluación (Consejo de Facultad de Ingeniería).

Problema recorrido de ida y vuelta

Sea C una cuadrícula de tamaño $n \times n$, compuesta por n filas y n columnas. En esta cuadrícula, se buscan recorridos que conecten distintas celdas, donde los movimientos permitidos son únicamente en direcciones horizontales o verticales.

Cada celda de la cuadrícula puede tener uno de los siguientes tres estados: libre, ocupado o prohibido. Los recorridos solo pueden incluir celdas que no estén en estado prohibido. Las celdas se identifican por sus coordenadas, fila f y columna c, siendo la celda $(1, 1)$ la ubicada en la esquina superior izquierda.

Para simplificar el tratamiento de los casos límite, se asume la existencia de filas 0 y $n + 1$, así como columnas 0 y $n + 1$, cuyas celdas son consideradas automáticamente como prohibidas.

(a) Se debe encontrar un recorrido donde los movimientos horizontales sean solo a la derecha y los verticales solo hacia abajo, desde la celda $(1, 1)$ hasta la celda (n, n) y que pase por la mayor cantidad posible de celdas ocupadas o, si a causa de la posición de las celdas prohibidas ese recorrido no existe, indicarlo.

Se debe diseñar un algoritmo que resuelva el problema cuyo tiempo de ejecución sea $O(n^2)$. Para hacer esto, puede comenzar considerando el problema genérico en el que el recorrido empieza en la celda (f, c) y $\mathrm{OPT}(f, c)$ representa la máxima cantidad de celdas ocupadas por las que se puede pasar en un recorrido desde (f, c) hasta (n, n).

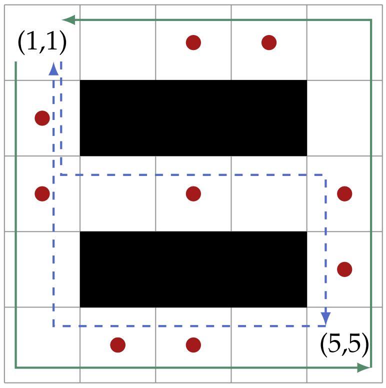
  I. Formule una relación de recurrencia para $\mathrm{OPT}(f, c)$ y justifique su corrección.

  II. Escriba un algoritmo que calcule $\mathrm{OPT}(1, 1)$. El algoritmo debe ser iterativo y aplicar la relación de recurrencia anterior.

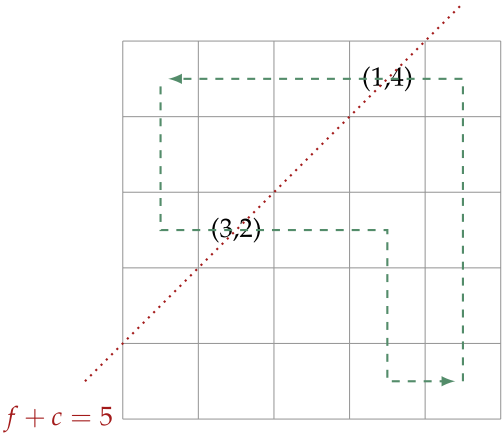
  III. Escriba un algoritmo que obtenga el recorrido desde $(1, 1)$ hasta (n, n) que pasa por una máxima cantidad de celdas ocupadas moviéndose de la forma indicada más arriba.

(b) En esta parte, el recorrido debe ir desde $(1, 1)$ hasta (n, n) como en la parte anterior, pero después se debe volver a $(1, 1)$. Al volver, solo se puede mover hacia la izquierda y hacia arriba. El objetivo sigue siendo pasar por la mayor cantidad de celdas ocupadas. Es posible pasar por una misma celda tanto en la ida como en la vuelta, pero en el caso que sea una celda ocupada solo se la cuenta una vez.

Como forma de comenzar a pensar el problema, considere el siguiente algoritmo que **no** lo resuelve correctamente:
Se encuentra un recorrido óptimo de ida desde $(1,1)$ hasta (n,n) como en la parte anterior; luego, se marcan como libres las celdas de ese recorrido. Se encuentra otro recorrido óptimo desde $(1,1)$ hasta (n,n) en el tablero modificado; se lo invierte para que quede como un recorrido de vuelta (desde (n,n) hasta $(1,1)$) y se lo concatena con el primer recorrido.
En la figura se ve mediante un contrajemplo que este algoritmo no siempre encuentra el recorrido óptimo. Las celdas prohibidas están en negro y las ocupadas tienen un punto rojo. El recorrido obtenido por el algoritmo se ve en la línea azul discontinua. El camino de ida, $((1,1)–(3,1)–(3,5)–(5,5))$, pasa por cinco celdas ocupadas, y el de vuelta por otras dos, llegando a un total de siete (se vuelve a pasar por las celdas (2,1) y (3,1) que inicialmente estaban ocupadas, pero no se las vuelve a contar porque ya se pasó por ellas en el primer recorrido). Sin embargo, hay dos recorridos que pasan por ocho celdas ocupadas, uno de los cuales se muestra con línea verde continua (el otro pasa por las mismas celdas pero en el sentido opuesto).



Para seguir pensando en el problema, consideremos la fila y columna genérica f y c, respectivamente. Notemos que dada cualquier diagonal $f + c = d$, $2 \leq d \leq 2n$, un recorrido debe pasar dos veces por ella (puede ser que las dos veces pase por la misma celda).
En la figura se ve un recorrido que pasa por la diagonal $f + c = 5$ en las celdas $(3,2)$ y $(1,4)$.



I. Ahora, consideremos la siguiente generalización del problema en la que las celdas de inicio y final no son la misma, sino que están en diagonales que van de izquierda a derecha y de abajo hacia arriba. Como la suma $f + c$ es constante en las celdas de una diagonal, se trata de las celdas (f,c) y $(f-k, c+k)$, con $0 \leq k \leq \min\{n-c, f-1\}$. Note que k puede ser 0, lo que produce el caso particular en que las celdas de inicio y de fin son las mismas.
Entonces, en esta versión general del problema, el recorrido de ida es de (f,c) hasta (n,n), y el de vuelta desde (n,n) hasta $(f-k, c+k)$.
Avanzando en la resolución del problema inicial, piense y explique cuáles son los subproblemas que se deben considerar para resolver el problema de general con inicio en (f,c) y fin en $(f-k, c+k)$.
Considere que $\text{OPT}(f,c,k)$ representa la máxima cantidad de celdas ocupadas por las que se puede pasar con un recorrido de ida desde (f,c) hasta (n,n) y uno de vuelta desde (n,n) hasta $(f-k, c+k)$.
Formule una relación de recurrencia para $\text{OPT}(f,c,k)$ y justifique su corrección.

II. Finalmente, resuelva el problema planteado inicialmente, es decir, el recorrido de ida es desde $(1,1)$ hasta (n,n) y el de vuelta es desde (n,n) hasta $(1,1)$.
El orden del tiempo de ejecución debe ser polinomial; analícelo y muestre por qué lo es.

Algo que puede ayudar a entender el problema es compararlo con otro problema conocido, por ejemplo el primero del libro, el de los intervalos. Una instancia de ese problema es un conjunto de n intervalos. Nos ayuda tratar ese conjunto como una secuencia ordenada según los tiempos de finalización. Para resolver el problema consideramos un conjunto, una estructura, de subproblemas, cada uno de los cuales está definido por los primeros i intervalos de la secuencia, para i desde 0 (o 1) hasta n. El problema inicial es un caso particular de este conjunto. Además, nos hacemos una pregunta acerca de la solución tal que cada posible respuesta determine cuál subproblema se debe tener resuelto. Si caracterizamos la solución en orden decreciente de sus índices, como $(s_h, s_{h-1}, \ldots, s_1)$ la pregunta es acerca de s_h, y es si s_h es el intervalo i. Si la respuesta fuera que no, entonces se debe resolver el subproblema de los primeros $i-1$ intervalos. Si la respuesta fuera que sí, entonces se debe resolver el subproblema de los primeros p_i intervalos (p_i es el último intervalo que no se solapa con i). Los dos subproblemas son los que luego se incluyen en la relación de recurrencia.

El problema de la parte (a) es encontrar un recorrido óptimo desde la celda $(1,1)$ hasta la celda (n,n). Igual que en el problema de los intervalos, se considera un conjunto de subproblemas, cada uno de los cuales está definido por una celda (f,c) genérica, y consiste en encontrar un recorrido óptimo desde (f,c) hasta la celda (n,n). Aquí también el problema inicial es un caso particular de este conjunto de subproblemas. Esta estructura de problemas es adecuada porque el recorrido óptimo desde $(1,1)$ hasta (n,n) incluye, para cada una de las celdas (f,c) que lo componen, el recorrido óptimo desde (f,c) hasta (n,n). O sea, incluye la solución del subproblema definido por (f,c). Para hacer la pregunta que determine los subproblemas que se deben tener resueltos, podemos caracterizar la solución como un recorrido que empieza en (f,c) y sigue con

$$(f',c') \to (f'',c'') \to \ldots \to (n,n).$$

Aquí debemos pensar cuál es la pregunta correspondiente a la que en el problema de los intervalos era saber algo acerca de s_h. Cada posible respuesta determina un subproblema que se incluye en la relación de recurrencia.

Después que hayamos resuelto el problema de la parte (a) pasamos al de la parte (b). Ahora el recorrido va desde $(1,1)$ hasta (n,n) y vuelve hasta $(1,1)$. Podríamos, igual que en el problema anterior, considerar que cada subproblema esté definido por una celda (f,c), y ese subproblema sería encontrar un recorrido óptimo que vaya desde (f,c) hasta (n,n) y vuelva a (f,c). Pero esto no es adecuado, porque si R es el recorrido óptimo para el problema $(1,1)$ y (f,c) es una de sus celdas, en general R no contiene el recorrido óptimo del subproblema (f,c) (porque en general R no pasa dos veces por la misma celda). Por eso es que se amplía la estructura de subproblemas. En esta nueva estructura un subproblema está definido por una celda de inicio y una celda de fin. Esta estructura es adecuada porque si R es un recorrido óptimo para el subproblema definido por las celdas (f,c) y $(f-k, c+k)$ e incluye las celdas (f',c') y $(f'-k', c'+k')$, entonces R contiene el reorrrido óptimo entre estas últimas celdas, o sea, la solución del problema definido por las celdas $(f'c')$ y $(f'-k', c'+k')$. El problema inicial, recorrido de ida y vuelta desde $(1,1)$, es un caso particular de este conjunto de subproblemas en el que las celdas de inicio y de fin son la misma, $(1,1)$.

Al problema definido por las celdas (f,c) y $(f-k,c+k)$ que están en una misma diagonal lo denotamos mediante la fila f y la columna c de la celda de inicio, y la distancia k entre las celdas de inicio y fin, (f,c,k). De manera similar a lo que vimos para la parte (a), podemos caracterizar la solución para el problema (f,c,k) como un recorrido que empieza en (f,c), sigue con un camino de ida hasta (n,n) y uno de vuelta,

$$\begin{array}{ll}
(f'_i, c'_i) \to \; (f''_i, c'_i) \to \; \ldots (f^{(h)}_i, c^{(h)}_i) \searrow & \\
 & (n,n) \\
(f'_v, c'_v) \leftarrow \; (f''_v, c'_v) \leftarrow \; \ldots (f^{(h)}_v, c^{(h)}_v) \swarrow &
\end{array}$$

y termina en $(f-k, c+k)$.

Hay que notar que en cada paso en el recorrido de ida se incrementa uno, o bien la fila o bien la columna, por lo que se pasa a la diagonal siguiente. Y en el recorrido de vuelta se decrementa uno, o bien la fila o bien la columna, por lo que se pasa a la diagonal anterior. Teniendo esto en cuenta, se puede ver que el par de celdas $(f'_i, c'_i), (f'_v, c'_v)$ pertenecen a la diagonal siguiente a la de $(f,c), (f-k, c+k)$, y que $(f''_i, c''_i), (f''_v, c''_v)$ pertenecen a la diagonal siguiente, y lo mismo se cumple con los siguientes pares.

En el problema de los intervalos nos preguntamos acerca de los posibles valores de s_h y nos interesaban dos. En el problema de la parte (a), ¿nos preguntamos acerca de los posibles valores de (f', c'), y también nos interesan dos? En el problema de la parte (b), teniendo en cuenta que los subproblemas están definidos por dos celdas de la misma diagonal, ¿qué podemos preguntar acerca de la solución cuya respuesta determine un subproblema? Hay varias respuestas, tal vez más de dos, cada una de las cuales determina un subproblema. Es conveniente preguntarse que condición debe cumplirse entre (f, c) y (f'_i, c'_i), y entre $(f-k, c+k)$ y (f'_v, c'_v).

Otras sugerencias

- En la relación de recurrencia es posible que convenga tratar de manera diferente el caso $k = 0$, o sea, cuando las celdas de inicio y fi son la misma.
- Los estados de las celdas de la cuadrícula C, la instancia del problema, cuyos valores son libre, ocupado o prohibido, puede convenir modelarlos mediante números, por ejemplo 0, 1 e $-\infty$, respectivamente. Si no es posible realizar un recorrido desde (f, c) hasta $(f-k, c+k)$, de manera consistente con lo anterior se le puede asignar $-\infty$ a $\mathrm{OPT}(f, c, k)$.
- Si se usa la sugerencia de agregar filas y columnas fuera de los bordes y asignarles estado prohibido, entonces no hay recorrido si una de las celdas está fuera de los bordes. Para esos casos, al asignar casos base de OPT alcanza con decir que si (f, c) o $(f-k, c+k)$ están fuera de los bordes entonces $\mathrm{OPT}(f, c, k)$ es $-\infty$ (si se sigue la sugerencia del punto anterior).