Ejercicios y problemas (ALGBIO)

Sandra Mingo Ramírez

2024/25

Ejercicio 1: Sort the following functions according to their growth:
$n, \sqrt{n}, n^{1.5}, n^2, n \log n, n \log \log n, n \log^2 n, 2/n, 2^n, 2^{n/2}, 37, n^2 \log n, n^3$

Para ordenar las funciones utilizando la notación O, hay que ver qué ocurre cuando n tiende a infinito. De forma ascendente:
$2/n, 37, \sqrt{n}, n, n \log \log n, n \log n, n \log^2 n, n^{1.5}, n^2, n^2 \log n, n^3, 2^{n/2}, 2^n$

Ejercicio 2: **Python** A classical example of a Divide and Conquer algorithm is binary search over ordered tables, which is essentially done according to the following pseudocode:

```
def bin_search(key, l_ints):
   m = len(l_ints)//2
   if key == l_ints[m]:
      return m
   elif key < l_ints[m]:
      search key on l_ints ip to index m-1 # left table search
   else:
      search key on l_ints from index m+1 # right table search
```

Expand the pseudocode into a correct recursive Python function.

```
def bin_search(key, l_ints):
   m = len(l_ints)//2
   if key == l_ints[m]:
      return m
   elif key < l_ints[m]:
      bin_search(key, l_ints[:m])
   else:
      bin_search(key, l_ints[m + 1:])
```

Ejercicio 3: Identify a proper key operation for `bin_search`. How many key comparisons are performed at most on the table [1, 2, 3, 4, 5, 6, 7] in successful searches? And in unsuccessful ones?

La operación clave es la comparación `key == l_ints`. En el mejor de los casos, solo se ejecuta una vez (si buscamos el 4), pero en el peor de los casos, 3 veces (y la tabla se termina sin haber encontrado el número).

Ejercicio 4: In general, how many unsuccessful searches will be performed at most on a table [1, 2, 3, ..., 2**n-1]? And successful ones?

Son necesarias n comparaciones, que es igual al $\log 2^n - 1$ o el logaritmo del tamaño de la tabla.

Ejercicio 5: **Fibonacci numbers and rabbit biology** Suppose that bunnies become adults in a month and that a pair of adult rabbits, regardless of sex, produce a pair of bunnies a month. If two young shipwrecked rabbits arrive on a deserted island, how many pairs of rabbits willl there be on the island after N months? For example, after two months there would be two pairs, one of bunnies and one of adult rabbits.

$$f_0 = f_1 = 1; f_n = f_{n-1} + f_{n-2}$$

Ejercicio 6: **Python** The Fibonacci numbers F_n are recursively defined as $F_0 = F_1 = 1, F_n = F_{n-1} + F_{n-2}$ for $N \geq 2$. Write a recursive Python function `fibonacci_rec(n)` to compute them.

```
def fibonacci(n):
   if n==0 or n==1:
      return 1
   return fibonacci(n-1) + fibonacci(n-2)
```

Ejercicio 7: **Python** Write an iterative version of `fibonacci(n)` which at each step saves the just computed Fibonacci numbers so that no recursive calls are needed. *Hint: use first a dict for this; then try to remove it.*

```
def fibonacci_iterative(n):
   d = {}
   d[0] = 1
   d[1] = 1

   for k in range(2, n+1):
      d[k] = d[k-1] + d[k-2]

   return d[n]
```

La alternativa sería utilizar un contador en lugar de un diccionario, lo cual sería más rápido.

Ejercicio 8: **Python** Modify our Hanoi code to write a function `hanoi_count(n_disks, a = 1, b = 2, c = 3)` that returns the required number of single disk movements.

Hacemos una lista con el número de discos y los movimientos necesarios:

- 1; 1
- 2; 3
- 3; 3+1+3=7
- 4; 7+1+7=15

Experimentalmente, vemos que los movimientos son $2^n - 1$.

Ejercicio 9: We have checked numerically that the number $n_{moves}(n)$ of disk moves the Hanoi algorithm makes over n disks is $2^n - 1$ but we don't have a proof on this. Try to write down one starting with the recurrence

$$N_{moves}(n) = 1 + 2 \times n_{moves}(n-1), n_{moves}(1) = 1$$

and applying induction on n.

Vemos que $m(4) = m(3) + 1 + m(3) = 1 + 2m(3)$, que de forma general se traduce en $m(n) = m(n-1) + 1 + m(n-1) = 1 + 2m(n-1)$. Esto es una ecuación recurrente.

$$m(n) = 1 + 2m(n-1) = 1 + 2(1 + 2m(n-2)) = 1 + 2 + 2^2 m(n-2) = 1 + 2 + 2^2 + 2^3 m(n-3)$$

Esto se debe ir desglosando hasta que lleguemos a $m(1) = 1$. Matemáticamente, $1 = m(n - (n-1))$.

$$m(n) = 1 + 2 + 2^2 + 2^3 + \ldots + 2^{n-2} + 2^{n-1}$$

Ejercicio 10: Find by hand the minimum distance between the strings `bahamas` and `bananas`. Find also a longest common subsequence.

	o	b	a	h	a	m	a	s
o	0	0	0	0	0	0	0	0
b	0	**1**	1	1	1	1	1	1
a	0	1	**2**	2	**2**	2	**2**	2
n	0	1	2	2	2	2	2	2
a	0	1	**2**	2	**3**	3	**3**	3
n	0	1	2	2	3	3	3	3
a	0	1	**2**	2	**3**	3	**4**	4
s	0	1	2	2	3	3	4	**5**

Creamos la tabla de edición. Desde la última celda, reconstruimos la matriz y vemos que la palabra es ba-a-as. Si los caracteres son iguales, nos movemos en diagonal (negrita en la tabla), y si son distintos, vemos las celdas de arriba o izquierda y le sumamos 1.

Ejercicio 11: Write down the adjacency list

Ejercicio 12: Given the 3-spectra:

$$S_{1,3} = [ATG, TGG, TGC, GTG, GGC, GCA, GCG, CGT]$$

$$S_{2,3} = [ATG, GGG, GGT, GTA, GTG, TAT, TGG]$$

find sequences s_i whose spectra are S_i writing down this problem as that of finding an adequate Hamiltonian path in a suitable graph and finding it by inspection.

Empezamos por el espectro 2.

- 1 - 7
- 2 - 3
- 3 - 4 - 5
- 4 - 6
- 5 - 7
- 6 - 1
- 7 - 2 - 3

Con esto, empezamos a pintar el gráfico de los caminos posibles. Hay que pasar por cada nodo una única vez. Vemos que un camino posible es 5 - 7 - 2 - 3 - 4 - 6 - 1. Ahora queda reconstruir la secuencia: GTGGGTATG.

Ejercicio 13: Now find sequences s_i whose spectra are S_i writing down a new problem of finding an adequate Eulerian path in a suitable graph and finding it from the graph's adjacency list.

Seguimos con el espectro 2 y su secuencia reconstruida y creamos la lista de adyacencia.

$$s_2 = [AT, TG, GG, GT, TA]$$

- 1 - 2
- 2 - 3
- 3 - 3 - 4
- 4 - 2 - 5

- 5 - 1

Volvemos a crear el dibujo con los posibles gráficos. Para ver el camino euleriano de forma algorítmica, hay que calcular los indegrees y outdegrees.

- 1 indegree, 1 outdegree: 1 - 2
- 2 indegree, 1 outdegree: 2 - 3
- 2 indegree, 2 outdegree: 3 - 3 - 4
- 1 indegree, 2 outdegree: 4 - 2 - 5
- 1 indegree, 1 outdegree: 5 - 1

Viendo esto, vemos que 1, 3 y 5 van a ser vértices centrales. Del 4 salimos dos veces pero llegamos 1 vez, por lo que debe ser el inicio del grafo. De 2 salimos una vez, pero llegamos dos, por lo que debe ser el final. Así, el camino resultante es 4-2-3-3-4-5-1-2 y la secuencia $S = [GTGGGTATG]$.

Ejercicio 14: A child's game is to try to draw the house below without lifting the pen from the sheet. Interpret this as a problem of finding an adequate Eulerian path in an undirected graph and find a such path starting at the lowest numbered node where it may start and "taking a walk" on the graph's adjacency list while providing adequate detail.

Vamos a crear la lista de adyacencia y su degree.

- 2 degree: 1 - 2 - 3
- 4 degree: 2 - 1 - 3 - 4 - 5
- 4 degree: 3 - 1 - 2 - 4 - 5
- 3 degree: 4 - 2 - 3 - 5
- 3 degree: 5 - 2 - 3 - 4

Vemos que todos los nodos salvo 4 y 5 tienen un valor par de degree, por lo que uno de ellos es el inicio y otro el final. Supongamos que empezamos por el 4, un posible camino sería: 4-2-1-3-2-5-3-4-5.

Ejercicio 15: We have solved in the lectures the problem of giving optimal change of 7 maravedís with coins of 1, 3, 4, 5 maravedís. Using the dynamic programming matrix, try to give a procedure to determine the coins that will enter in the optimal change.

	0	1	2	3	4	5	6	7
1	0	1	2	3	4	5	6	7
3	0	1	2	1	2	3	2	3
4	0	1	2	1	1	2	2	2
5	0	1	2	1	1	1	2	2

La última celda indica el número de monedas a usar, y se pueden usar 2 monedas. Ahora hay que deshacer la matriz. Partimos de la celda de arriba, por lo que no se usan monedas de 5. El siguiente movimiento está en esa misma fila, por lo que se emplea una moneda de 4. A continuación se sube, y nos desplazamos a la izquierda, usando una moneda de 3. Como hemos llegado a 0, ahí se termina.

Ejercicio 16: The president of the United States has a leaking problem: one among his 100 person staff keeps leaking stories to the press. To find him/her out, his Chief Computing Officer tells him to give different stories to a number of different staffers and check whether the information ends up in the news until they find out (they know that he or she will leak any story he or she is given). Since coming up with

stories is hard, they want to minimize the number that they use. Devise a strategy to handle stories and to find out this minimum number.

Utilizando búsqueda binaria, con 1 noticia pasamos de 100 candidatos a 50, luego a 25, 13, 7, 4, 2 y 1. Así, sólo serían necesarias 7 noticias.