

Exàmens de PRG - Problemes del Tema 1

Recursió

Curs 2018/19

P1 - Curs 18/19: 4 punts

Donats un array d'enters a i un enter k , $k \geq a.length$, es diu que a és *un capicua de clau k* si la diferència en valor absolut entre les components primera i última és menor que k , menor que $k - 1$ entre les components segona i penúltima, menor que $k - 2$ entre la tercera i l'antepenúltima, i així successivament. S'entén que l'array buit o amb una sola component són capicues, per a qualsevol clau. Es vol un mètode recursiu que comprovi si a és un capicua de clau k . Per exemple,

- per a l'array $\{20, 15, 14, 32, 10, 7, 22\}$ i la clau 10, el mètode ha de retornar **true** ja que:
 - $|20 - 22| < 10$,
 - $|15 - 7| < 9$,
 - $|14 - 10| < 8$ i
 - trivialment, $|32 - 32| < 7$.
- per a l'array $\{20, 15, 14, 10, 7, 22\}$ i la clau 10, el mètode també ha de retornar **true** ja que:
 - $|20 - 22| < 10$,
 - $|15 - 7| < 9$,
 - $|14 - 10| < 8$.
- per als arrays anteriors i la clau 9 el mètode ha de retornar **false**, atès que:
 - $|20 - 22| < 9$, però
 - $|15 - 7|$ no és menor que 8.

Es demana:

- (0.75 punts) Perfil del mètode recursiu, amb els paràmetres adequats per a resoldre recursivament el problema, i precondition relativa a aquests paràmetres.
- (1.25 punts) Cas base i cas general de la recursió.
- (1.50 punts) Implementació en Java del mètode recursiu.
- (0.5 punts) Crida inicial al mètode per a comprovar si un cert array **data** és o no capicua de clau **k**.

RecP1 - Curs 18/19: 4 punts

Donat un enter $n > 0$, escriu un mètode recursiu que mostri per l'eixida estàndard les xifres de n en sentit invers, seguides de les mateixes xifres en el sentit en que apareixen en n . Per exemple, si n és 4873, cal escriure 37844873; si n és 48, cal escriure 8448; si n és 4, cal escriure 44.

Es demana:

- (0.75 punts) Perfil del mètode amb la seua precondition.
- (1.25 punts) Cas base i cas general.
- (2 punts) Implementació en Java.

Curs 2017/18

P1 - Curs 17/18: 4 punts

Donats un array `v` de `String` i un nombre natural `n`, escriu un mètode **recursiu** que retorne quants elements de l'array `v` tenen una longitud `n`.

Per exemple, si l'array `v` continguera `{"barco", "autobus", "tren", "moto", "bici"}`, i `n` fóra 4, llavors el mètode retornaria 3. Si l'array fóra el mateix, i `n` fóra 5, llavors el mètode retornaria 1. Per al mateix array `v`, i `n` igual a 3, el mètode retornaria 0.

Es demana:

- a) (0.75 punts) Perfil del mètode, amb els paràmetres adequats per a resoldre recursivament el problema, i preconditionió relativa a aquests paràmetres.
- b) (1.25 punts) Cas base i cas general.
- c) (1.50 punts) Implementació en Java.
- d) (0.50 punts) Crida inicial perquè es realitzi el càlcul sobre tot l'array.

RecP1 - Curs 17/18: 4 punts

Siga `a` un array de `double`, on les seues components representen valors de la coordenada Y d'una enumeració de punts pertanyents a una recta de pendent positiva, de manera que l'array està ordenat ascendentment.

S'ha d'escriure un mètode recursiu que cerque el punt de tall de la recta amb l'eix X, o que retorne -1 si el punt de tall no es troba entre els d'`a`.

Per exemple: si l'array és `{-7.4, -1.3, 0.0, 1.8, 2.3, 3.6}`, el mètode ha de retornar 2. Si l'array és `{-7.4, -1.3, 1.8, 2.6, 3.6}`, el mètode ha de retornar -1.

Es demana:

- a) (0.75 punts) Perfil del mètode, amb els paràmetres adequats per a resoldre recursivament el problema, i preconditionió relativa a aquests paràmetres.
- b) (1.25 punts) Cas base i cas general.
- c) (1.50 punts) Implementació en Java.
- d) (0.50 punts) Crida inicial perquè, donat un cert array `a`, es realitzi el càlcul sobre tot l'array.

Curs 2016/17

P1 - Curs 16/17: 4 punts

Donat un array `a` de `int` i un enter `x`, escriu un mètode **recursiu** que torne quants múltiples de `x` hi ha a l'array `a`.

Es demana:

- a) (0.75 punts) Perfil del mètode, amb els paràmetres adequats per a resoldre recursivament el problema.
- b) (1.25 punts) Cas base i cas general.
- c) (1.50 punts) Implementació en Java.
- d) (0.50 punts) Crida inicial perquè es realitzi el càlcul sobre tot l'array.

RecP1 - Curs 16/17: 3 punts

Un número triangular és aquell que pot recomposar-se en la forma d'un triangle equilàter (considerarem que el primer número triangular és el 0). Els primers 10 números triangulars són: 0, 1, 3, 6, 10, 15, 21, 28, 36, 45.

Cada número triangular es pot calcular com:

$$t(n) = \begin{cases} 0 & n = 0 \\ n + t(n-1) & n > 0 \end{cases}$$

On $t(n)$ és l'enèsim número triangular i $t(n-1)$ el seu anterior.

Es vol implementar, usant recursió, un algorisme que permeti instanciar un array d'un cert tamany n , i que continga els n primers termes d'aquesta successió dels números triangulars.

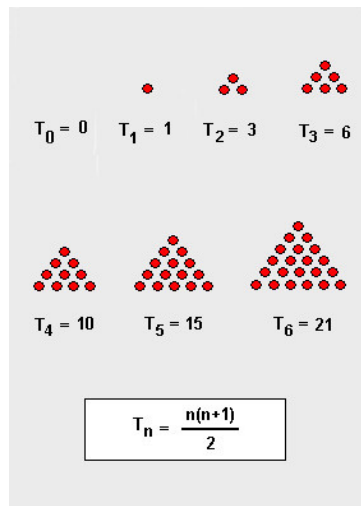


Figura 1: Representació gràfica dels 7 primers números triangulars

Es demana implementar els següents mètodes, per als que a més s'hauran d'especificar les precondicions necessàries:

- Un primer mètode, llançadora, no recursiu, que reba un únic argument, el número n , que haurà de ser un valor enter positiu. Aquest mètode haurà d'instanciar l'array al tamany adequat, invocar al segon mètode (encarregat de reomplir l'array), i retornar l'array.
- Un segon mètode, recursiu, que reba dos arguments: l' array de tamany n i un altre argument, necessari per a implementar el disseny recursiu. En aquest mètode s'hauran de calcular els valors de la successió, i magatzemar-los en les posicions que corresponga en l'array.

Per exemple, la crida al mètode llançadora amb $n = 5$ haurà de retornar l'array $\{0, 1, 3, 6, 10\}$, mentre que amb $n = 1$ haurà de retornar l'array $\{0\}$.

Curs 2015/16

P1 - Curs 15/16: 4 punts

Donat un array **a** de **int** ordenat ascendentment i un enter **x**, escriu un mètode **recursiu** que calcule la suma dels elements menors que **x** que hi ha a **a**. Per exemple, per **a** = $\{2, 2, 7, 8, 10, 10, 12, 23, 34\}$ i **x** = 10 ha de tornar 19; per al mateix array i **x** = 1 ha de tornar 0. El mètode ha d'evitar fer comparacions supèrflues d'elements de **a** amb **x**.

Es demana:

- (0.75 punts) Perfil del mètode, amb els paràmetres adequats per a resoldre recursivament el problema.
- (1.25 punts) Cas base i cas general.
- (1.5 punts) Implementació en Java.
- (0.5 punts) Crida inicial perquè es realitzi el càlcul sobre tot l'array.

RecP1 - Curs 15/16: 4 punts

Donat un array **a** de **int** on **a.length** ≥ 1 , escriure un mètode **recursiu** que determine si els elements de l'array representen una successió de Fibonacci, és a dir, si el valor d'un element correspon a la suma dels dos elements immediatament anteriors (sent els dos primers elements 0 i 1), tal com es mostra en el següent exemple: $\{0, 1, 1, 2, 3, 5, 8, 13, 21, \dots\}$.

Es demana:

- (0.75 punts) Perfil del mètode, amb els paràmetres adequats per a resoldre recursivament el problema.
- (1.25 punts) Cas base i cas general.
- (1.5 punts) Implementació en Java.
- (0.5 punts) Crida inicial per a que determine si es compleix la successió sobre tot l'array.

Curs 2014/15

P1 - Curs 14/15: 3 punts

Donat un array **a** de **int** amb almenys un element, escriure un mètode **recursiu** que comprovi si tots els valors de l'array són parells i estan emmagatzemats en ordre creixent.

Es demana:

- a) (0.5 punts) Perfil del mètode, amb els paràmetres adequats per tal de resoldre recursivament el problema.
- b) (1.2 punts) Cas base i cas general.
- c) (1 punt) Implementació en Java.
- d) (0.3 punts) Crida inicial perquè es verifiqui la propietat sobre tot l'array.

Curs 2013/14

P1 - Curs 13/14: 3 punts

Donat un array **a** de **int**, escriure un mètode **recursiu** que sumi els números capicua des dels extrems de l'array, definits per l'interval $[ini..fi]$ ($0 \leq ini, fi < a.length$), cap al centre fins sumar-los tots o trobar algun parell de números que no siguin iguals. En cas que l'array tinga una quantitat senar d'elements, l'element central de l'interval només ha de sumar-se una vegada. Si l'array no té cap element capicua en els extrems de l'interval, el resultat és zero.

Alguns exemples:

- per a l'array **a** = {1,4,1,2,4,1}, el resultat de la suma és 10 (1+1+4+4),
- per a l'array **a** = {1,4,2,2,4,1}, el resultat de la suma és 14 (1+1+4+4+2+2),
- per a l'array **a** = {1,4,1,4,1}, el resultat de la suma és 11 (1+1+4+4+1),
- per a l'array **a** = {1,4,0,0,1}, el resultat de la suma és 2 (1+1),
- per a l'array **a** = {8,4,0,0,1}, el resultat de la suma és 0.

Es demana:

- a) (0.5 punts) Perfil del mètode, amb els paràmetres adequats per tal de resoldre recursivament el problema.
- b) (1.2 punts) Cas base i cas general.
- c) (1 punt) Implementació en Java.
- d) (0.3 punts) Crida inicial.

RecP1 - Curs 13/14: 3 punts

Es demana: dissenyar un mètode **recursiu** que determine si cert array d'enters **v** conté els elements inicials d'una successió de Fibonacci. Per a això l'array es proporciona amb almenys 3 elements. Cal recordar que la successió de Fibonacci és la següent: 0,1,1,2,3,5,8 ... És a dir, el terme **n**-èsim és **n** si **n** = 0 o **n** = 1; en un altre cas, es calcula com la suma dels dos termes anteriors ((**n**-1)-èsim i (**n**-2)-èsim).

- (a) (2.5 punts). Escriure el mètode d'acord a les consideracions anteriors.
- (b) (0.5 punts). D'acord a la implementació del mètode anterior, indicar la crida inicial perquè es verifiqui la propietat sobre tot l'array.

Curs 2012/13

P1 - Curs 12/13: 4 punts

Donat un array `a` d'`int` i un enter `m`, escriure un mètode **recursiu** que comprovi si existeix una parella `a[i]` i `a[f]`, $0 \leq i \leq f < a.length$, de components *simètriques* (la distància d'`i` a 0 és la mateixa que la distància de `f` a `a.length - 1`) que sumen `m`. Si existeix la parella, ha de tornar l'índex en el que es troba (l'índex `i`, el més baix de la parella), `-1` en cas contrari.

Per exemple, per a `m = 4` i `a = {1,4,5,9,6,0,-8}` el mètode en la crida inicial que s'extenga sobre tot l'array ha de tornar 1, per a `m = 18` i `a = {1,4,5,9,6,0,2}` ha de tornar 3, per a `m = 25` i `a = {1,3,2,5,4,6}` ha de tornar `-1`.

Es demana:

- Perfil del mètode, amb els paràmetres adequats per tal de resoldre recursivament el problema.
- Cas base i cas general.
- Implementació en Java.
- Crida inicial.

RecP1 - Curs 12/13: 2.5 punts

Per a determinar si cert nombre enter no negatiu `n` pot estar expressat en una base determinada `b` ($2 \leq b \leq 10$), és suficient que tots els dígit del nombre tinguin un valor estrictament menor que la base `b`. Per exemple, el nombre 453123 pot representar un valor en base 6, 7, 8, 9 i 10 ja que tots els seus dígit són estrictament inferiors als valors d'aquestes possibles bases.

Es demana: implementar en Java un mètode **recursiu** que, donats `n` i `b`, resolga el problema plantejat, especificant els casos base i general de la recursió.

RecP1 - Curs 12/13: 2.0 punts

Donat un array `a` de valors reals i cert valor real `x`, es vol determinar, **recursivament**, el nombre d'elements d'`a`, des de la posició `pos` inclosa fins a la final (`a[pos..a.length - 1]`), que tinguin valor més gran que `x`. Per a això, partint del perfil:

```
public static int numMajors(double[] a, double x, int pos)
```

Es demana:

- Implementar el mètode **recursiu** demanat, especificant els casos base i general de la recursió.
- Escriure la crida inicial per obtenir el nombre d'elements més grans que `x` de tot un array.

Curs 2011/12

P1 - Curs 11/12: 2 punts

Escriu un mètode **recursiu** que reba com argument un valor enter no negatiu i escriga en l'eixida estàndard les seqüències descendent i ascendent de valors enters des del número inicial fins el zero. Cada seqüència en una línia diferent.

Per exemple, si el mètode sol·licitat s'executa amb el valor 14, aleshores escriurà per l'eixida estàndard el següent:

```
14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

Nota que els números de cada seqüència apareixen tots seguits en una mateixa línia.

P1 - Curs 11/12: 2 punts

Escriu un mètode **recursiu** que, donats un array d'enters `v`, una posició inicial `ini` ≥ 0 , una posició final `fi` $< v.length$ i un valor enter `x`, determine si la suma dels elements simètrics és `x`, és a dir, els extrems dels subarrays formats per les posicions en l'interval `[ini + i, fi - i]` per a $0 \leq i \leq (ini + fi) / 2$ compleixen que `v[ini + i] + v[fi - i] == x`.

Per exemple:

Si `x = 10`, `ini = 1`, `fi = v.length - 2` i `v = {1,2,3,4,5,6,7,8,9}` retorna `true`.

Si `x = 10`, `ini = 1`, `fi = v.length - 2` i `v = {1,2,4,4,5,6,7,8,9}` retorna `false`.

Si `x = 6`, `ini = 0`, `fi = v.length - 1` i `v = {1,2,3,3,4,5}` retorna `true`.

RecP1 - Curs 11/12: 2 punts

Siga `a` un array d'int i `x` un int. **Es demana** un mètode **recursiu** que indique si els elements d'`a` formen una progressió geomètrica de raó `x`, és a dir, si cada component `a[i + 1]` de l'array val `a[i] * x`. Per exemple, per a `a = {3,6,12,24,48}` i `x = 2` el mètode ha de retornar `true`, per a `a = {3,6,12,33,48}` i `x = 2` el mètode ha de retornar `false`. Si sols hi ha un element, s'entén que és una progressió geomètrica siga quin siga `x`.

Indicar quina haurà de ser la primera crida.

RecP1 - Curs 11/12: 2 punts

Escriure un mètode **recursiu** que donat un enter `n` ≥ 0 , escriga en la sortida estàndard, i en la mateixa línia, els valors `-n -(n - 1) ... -2 -1 0 1 2 ... (n - 1) n`. Per exemple, per a `n = 3`, en la sortida s'ha d'escriure:

```
-3 -2 -1 0 1 2 3
```