PRG (E.T.S. d'Enginyeria Informàtica)
Curs 2015-2016
Pràctica 1. La recursió com estratègia de resolució de problemes: les Torres d'Hanoi

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València

Índex

1 Context i treball previ a la sessió de pràctiques

En el marc acadèmic, la present pràctica és la primera, corresponent al *"Tema 1. Recursió"* de l'assignatura. Els objectius principals que es pretenen aconseguir amb ella són dos: en primer lloc, que reforces i poses en pràctica els conceptes introduïts en les classes de teoria sobre estratègia i disseny recursiu; en segon lloc, que d'ara endavant sigues capaç de valorar per experiència pròpia l'útil que et pot resultar plantejar una estratègia recursiva per a resoldre un problema complex.

Per a aconseguir estos objectius, durant la sessió de pràctiques hauràs d'abordar i resoldre les distintes etapes del disseny recursiu d'un mètode Java que soluciona el problema de les Torres d'Hanoi, de gran pes específic en Programació. Per a realitzar estes activitats, i també per a ajudar-te i guiar-te en la seua resolució, hauràs d'interactuar amb una interfície gràfica que s'ha dissenyat expressament per a l'ocasió, l'executable de la qual el tens disponible en l'adreça

`Recursos: Laboratorio: Práctica 1` de la `PoliformaT` de `PRG`. Una vegada conclogues el disseny d'Hanoi, a manera d'autoavaluació de l'après, en `PoliformaT PRG: Exàmens` pots realitzar uns exercicis addicionals relacionats amb el treball realitzat sobre la pràctica. Aquest últim test és voluntari i no afecta a la nota de l'assignatura.

Per tot l'anterior, perquè aprofites al màxim la sessió de pràctiques, t'aconsellem que, abans, realitzes una lectura comprensiva d'aquest butlletí, et familiaritzes amb la interfície i, finalment, interactues amb ella per a resoldre el problema plantejat.

2 Raonament recursiu: conceptes bàsics

Els termes Inducció, Recursió o Inferència s'utilitzen indistintament i en diversos àmbits, des del quotidià al científic, per a denominar una mateixa estratègia de resolució de problemes: a partir d'un conjunt finit de solucions d'un problema per a casos senzills, es realitza una hipòtesi per a obtindre la solució general d'eixe problema, i.e. la solució aplicable a qualsevol dels seus casos o cas general.

Per a posar de manifest la naturalitat amb què els éssers humans usem l'estratègia inductiva, a propòsit o no, et plantegem ara un problema senzill que, segur, ja has resolt aplicant-la en més d'una ocasió: suposa que durant la realització d'un test psicotècnic, et donen la sèrie de valors `2`, `4`, `8`, `16`, `32`, `64` i et demanen determinar quin és el valor que segueix a l'últim (`64`). Després d'examinar un poc els valors que et donen, no dubtaries a respondre que al `64` li segueix el `128`; és més, podries continuar obtenint nous valors de la sèrie a partir del `128` sense majors problemes: `256`, `512`, `1024`, ...

Tampoc et resultaria molt difícil explicitar la relació que existeix entre qualsevol valor d'aquesta sèrie i l'anterior: $\forall\ i \geq 2$ l'`i`-èsim valor s'obté a partir de l'`i-1`-èsim per mitjà del càlcul $2 * 2^{i-1}$; observa que ho pots fer perquè és la fórmula que, conscientment o inconscientment, has aprés a partir dels valors donats i la que has aplicat per a poder calcular el següent a `64`, o a `128`. A partir d'ací és immediat veure que la funció, els valors de la qual es calculen d'esta manera, és la **exponencial** 2^i, perquè com ja saps $2^i = 2^1 * 2^{i-1}$.

Qüestió: comprova que has entés este exemple completant amb, almenys, dos valors més cada una de les sèries numèriques que apareixen a continuació; indica també per a cada una d'elles l'expressió general que apliques per a calcular el seu `i`-èsim valor a partir de l'anterior (o anteriors) i la funció matemàtica a què pertany el valor calculat. Finalment, decideix què seria més fàcil, si implementar un mètode iteratiu o un recursiu per a obtindre el `i`-èsim valor de qualsevol de les funcions obtingudes.

- `1, 1, 2, 6, 24.`
- `1, 3, 7, 15, 31, 63, 127.`
- `0, 1, 3, 6, 10, 15.`
- `1, 1, 2, 3, 5, 8.`

Altres problemes interessants que resolem utilitzant l'estratègia inductiva són l'aprenentatge de la llengua materna en els primers anys de la nostra vida i, en Matemàtiques, la definició de moltes funcions, entre altres factorial i Fibonacci, la de conjunts com els Naturals i la prova per Inducció d'una propietat donada. Analitzant-los com a processos d'aprenentatge ja experimentats, aquestos exemples ajuden a perfilar encara més les característiques del raonament inductiu:

1. La hipòtesi de solució del problema per al cas general s'expressa **explícitament** en termes de la(les) solució(ns) del mateix problema per a un cas(os) més senzill(s).

2. La hipòtesi de solució del problema per al cas general ha de ser validada o demostrada per mitjà d'un procés de prova, sense el qual no deixa de ser més que això, una hipòtesi. És precisament per esta característica que, moltes vegades, es denomina prova per inducció al raonament inductiu.

3. Com a avantatge fonamental, resulta més fàcil expressar la solució d'un problema complex inductivament, o en termes de la solució del mateix problema per a casos més senzills, que per enumeració dels passos que s'ha de seguir per a obtindre la solució.

3 Plantejament del problema

Precisament perquè d'ara endavant sigues capaç de plantejar estratègies recursives per a resoldre determinats problemes, apreciant l'útils que poden resultar enfront de les seues equivalents iteratives, en esta pràctica se't demana dissenyar un mètode Java que resolga el problema de *les Torres d'Hanoi*, l'enunciat del qual és el següent:

> Es disposa de tres torres (`origen`, `destí` i `auxiliar`) i un cert nombre de `discos` de diferents grandàries. Inicialment, tots els `discos` es troben en la torre `origen` apilats en forma decreixent segons el seu diàmetre. El problema consisteix a desplaçar-los a la torre `destí` utilitzant la torre `auxiliar` i seguint les regles següents: els discos s'han de desplaçar un a un i en cap moment pot haver-hi un disc de major diàmetre situat sobre un més xicotet.

4 Activitats per a la resolució recursiva del problema

Per a donar una solució recursiva al problema plantejat, i expressar-la en forma de mètode Java, has d'abordar i resoldre amb ajuda de la interfície gràfica disponible les quatre activitats que es detallen en les seccions d'este apartat, perquè a grosso modo cada una d'elles cobreix una etapa del disseny recursiu que has de realitzar. Com a pas previ, per a poder executar la interfície gràfica, has de ...

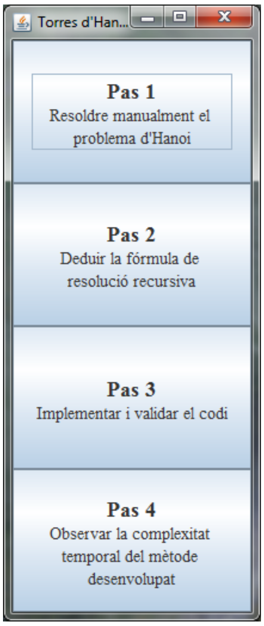
1. Obrir un terminal de consola i crear en `$HOME/DiscoW` un directori `prg`.

2. Crear un subdirectori `pract1` dins del directori `prg`.

3. Copiar en `pract1`, des de `Recursos: Laboratorio: Practica 1` de la `PoliformaT` de `PRG` el fitxer `HanoiGui.jar`, l'executable de la interfície.

4. Executar el comandament `java -jar HanoiGui.jar` des del terminal de la consola de Linux. Apareixerà llavors en la teua pantalla un menú gràfic com el que mostra la següent figura.

   Has de fer click en cada un dels seus `Pasos`, començant per l'`1` i acabant en el `4`, per a resoldre les activitats que es descriuen a continuació; fins que no estiga resolta una activitat, no has de passar a la següent.



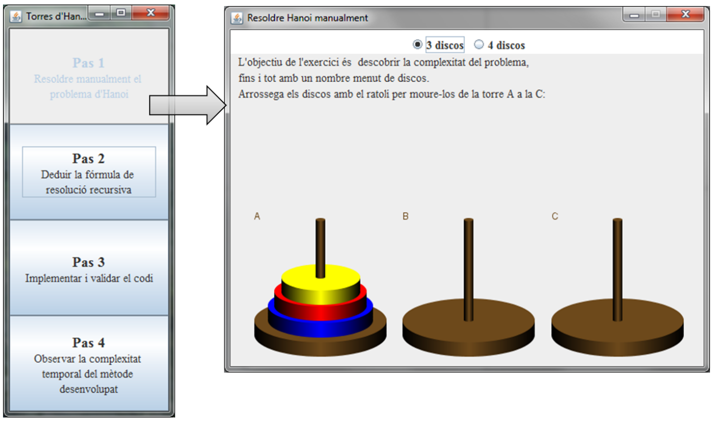
4.1 Activitat 1: Determinació de l'estructura i paràmetres del problema

La primera activitat que has de realitzar consisteix, bàsicament, en resoldre el `Pas 1` de la interfície gràfica, que apareix en la figura següent: resoldre manualment el problema de les Torres d'Hanoi per a `3` i `4` discos. Això et permetrà analitzar les característiques del problema i, en conseqüència, decidir si és prou complex com per a plantejar la seua solució recursiva en compte d'iterativament; també podràs entendre al realitzar esta activitat el paper, nombre i tipus dels paràmetres formals del mètode que dissenyaràs més tard, el tipus del seu resultat i, el que és més important, quin paràmetre del mètode és el que fa més fàcil o difícil la resolució del problema.

Si realment has aconseguit cobrir esta primera etapa del disseny recursiu a realitzar, no tindràs cap problema a resoldre les següents qüestions.

Qüestions sobre l'activitat 1: cara a dissenyar un mètode `hanoi`, que per exemple resolga en el teu lloc el `Pas 1` de la interfície,

- Quants paràmetres formals i de quins tipus Java inclouries en la seua capçalera? Indica quin nom donaries a cada un d'estos paràmetres perquè permeta identificar sense am-



bigüitat el que representa i quin d'ells estableix el grau de dificultat del problema de les Torres d'Hanoi.

- Quin seria el tipus dels resultats d'este mètode?

  Completa la següent taula per a fixar conceptes i nomenclatura:

naturalesa	tipus de retorn	nom	llista de paràmetres
`static`		`hanoi`	( )

- Com invocaries al mètode `hanoi`, la capçalera del qual acabes de definir, per a resoldre el `Pas 1` de la interfície?
- Encara que segur que series capaç de descriure la seqüència de moviments realitzada per a resoldre el problema de les Torres d'Hanoi per a `3` i `4 discos`, podries descriure la iteració que resol el problema per a qualsevol nombre de discos? I per a `1`, `2`, `3` i `4 discos`? A l'hora de respondre estes preguntes, suposa que en el joc d'instruccions de què disposes figura `moureDisc`, que mou el disc (de menor diàmetre) situat damunt del tot, en una torre `origen` donada, a una torre `destí` donada, sempre que això no incomplisca les regles d'Hanoi.

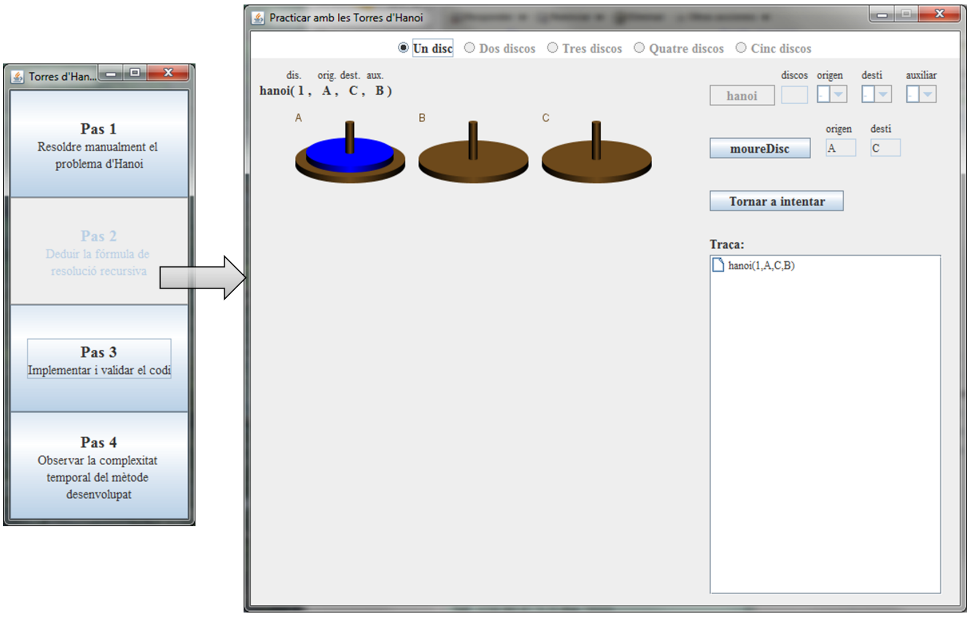
Si no eres capaç de resoldre l'última qüestió plantejada no et preocupes, és normal. Precisament una de les conclusions que has d'extraure després de realitzar aquesta activitat és que descriure pas a pas, iterativament, la solució de les Torres d'Hanoi és massa complex, resulta quasi impossible si s'afronta globalment seguint les regles establertes. Perquè et convences definitivament d'això, et mostrem ara una possible implementació iterativa del problema de les Torres d'Hanoi:

```
int h = discos; String a = origen, c = destí, b = auxiliar, k; int no = 1;
while ( no != 0 ) {
    while ( h != 2 ) { h--; k = b; b = c; c = k; no = 2 * no; }
    moureDisc(a, b); moureDisc(a, c); moureDisc(b, c);
    while ( (no % 2) != 0 ) { h++; k = a; a = b; b = k; no = no/2; }
    if ( no != 0 ) {
        moureDisc(a, b);
        k = a; a = c; c = b; b = k; no++;
    }
}
```

4.2 Activitat 2: Anàlisi de casos

La resolució de la primera activitat ha d'haver-te conduït a replantejar la teua estratègia inicial: en compte d'abordar la resolució general del problema tal com s'ha enunciat, lo més natural és intentar abordar la resolució del problema de les Torres d'Hanoi des del seu cas més senzill, en el que només cal moure 1 disc, per orde creixent de complexitat o nombre de discos a moure. D'esta manera podràs segurament apre<u>h</u>endre la fórmula que et permet expressar la seua solució per al següent cas en termes de la(les) que ja has obtingut per a un cas anterior, i.e. amb un disc menys; recorda que, de fet, este és l'avantatge més clar que presenta el raonament inductiu enfront del deductiu a l'hora de resoldre un problema prou complex.

Doncs bé, per a "obligar-te" a pensar recursivament, l'Activitat 2 consisteix bàsicament en resoldre el problema d'Hanoi per a `1`, `2`, `3`, `4` i `5 discos` usant el videojoc del `Pas 2` de la interfície gràfica, i que mostra la següent figura.



Convé que sàpigues ja que este videojoc no et permet passar d'un nivell `i` al següent ($1\leq i\leq 4$) fins que no ho hages resolt correctament utilitzant un dels seus dos comandaments: `moureDisc`, que només mou el disc de menor diàmetre de la torre `origen A` a la torre `destí C`, i `hanoi`, que executa automàticament el problema que has resolt en el nivell anterior (`i-1`) sempre que li passes els valors correctes de les torres `origen`, `destí` i `auxiliar`. Per això, perquè aprengues a utilitzar-ho ràpidament, et proposem resoldre les qüestions següents:

- Independentment del nombre de discos `i`, $2\leq i\leq 5$, que estiguen situats en la torre `origen`, què has de fer sempre amb els `i-1` discos que hi ha sobre el seu disc de major diàmetre per a poder moure'l a la torre `destí`? Expressa-ho per mitjà d'un dels dos comandaments del videojoc.
- Una vegada que tens en la torre `origen` només el disc de major diàmetre, a quina torre l'has de moure sempre? Expressa-ho per mitjà d'un dels dos comandaments del videojoc.
- Una vegada col·locat en el seu lloc el disc de major diàmetre, com situes els `i-1` restants damunt d'ell formant una torre d'Hanoi? De nou, expressa-ho per mitjà d'un dels dos comandaments del videojoc; al fer-ho, recorda on tens situats estos `i-1 discos` i quina torre està lliure en eixe moment.

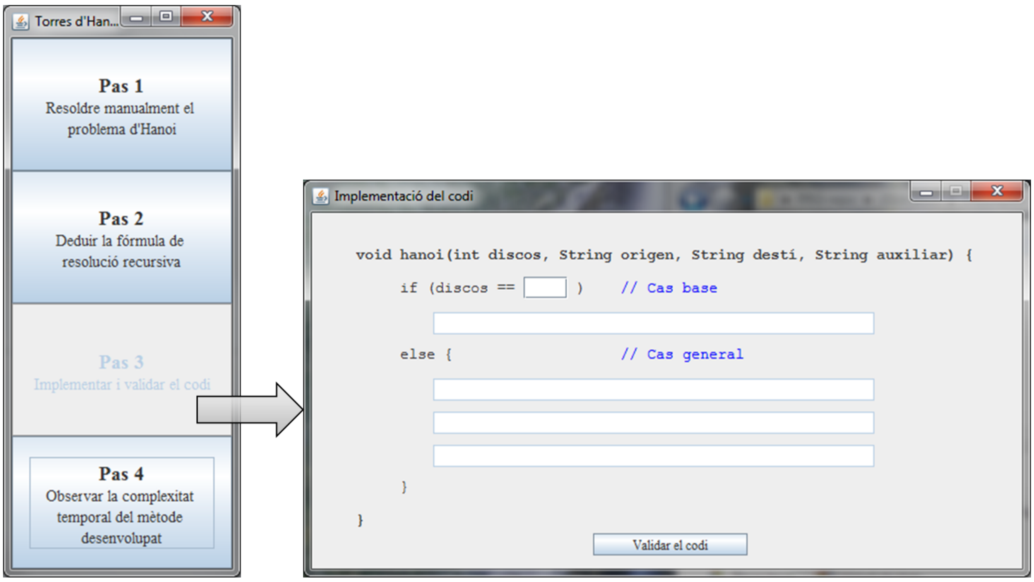
Si no aconsegueixes el nivell `5` del videojoc en un temps raonable, et recomanem que tornes a realitzar l'Activitat 1. En cas contrari, respon a les següents qüestions per a comprovar que has aprés la fórmula recursiva que resol el problema d'Hanoi; equivalentment, comprova si eres capaç de descriure els casos base i general de l'anàlisi de casos que has efectuat a l'usar el videojoc, així com les instruccions a ells associades.

Qüestions sobre l'activitat 2:

- Seleccionant successivament els botons `5 discos` i `4 discos` del videojoc, quina diferència s'observa en conjunt entre els estats de les torres `A`, `B` i `C`? Es repeteix esta diferència si el nombre de discos varia, successivament, entre `4` i `3`, `3` i `2`, i `2` i `1`? En cas afirmatiu, indica si la diferència observada és congruent amb l'estratègia recursiva que tens la intenció d'aplicar per a solucionar Hanoi.
- Repeteix el mateix procés que en la qüestió anterior amb el botó que indica el nombre de `discos` però fixa't esta vegada en què succeeix si, per a cada nombre de `discos` actiu, fas click en la carpeta que apareix en el requadre `Traça` situat en el cantó inferior dreta del videojoc. El que observes, és congruent amb l'estratègia recursiva que tens la intenció de proposar per a solucionar Hanoi? Reforça la teua idea que una solució iterativa d'Hanoi és una tasca impossible, o quasi?

4.3 Activitat 3: Transcripció Java de l'anàlisi de casos i validació del codi

Realitza el `Pas 3` de la interfície, que mostra la següent figura, per a expressar a Java l'estratègia o fórmula recursiva que has aprés en l'activitat 2. Valida després el codi que has escrit usant el botó que apareix davall d'ell; si el teu codi és correcte, i a manera de prova informal de correcció, traça amb ajuda de la interfície la seua execució quan és invocat per a un nombre de `discos` igual a `3`.



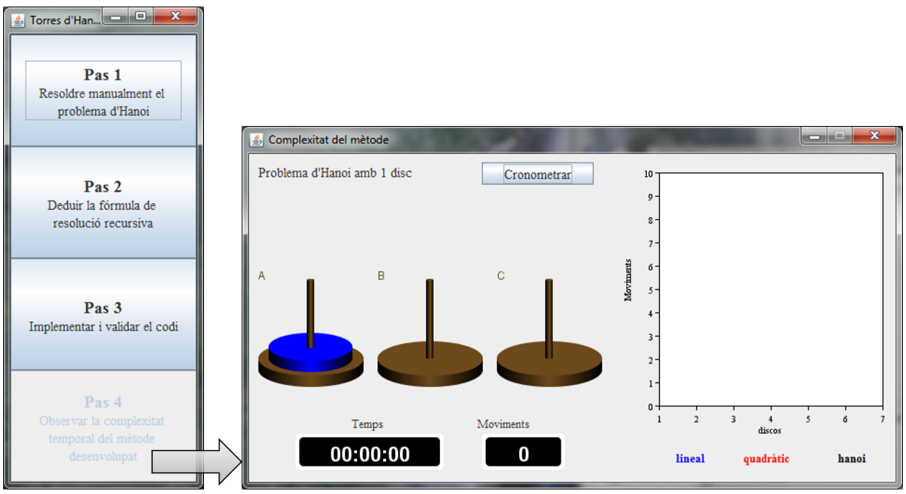
Si has observat amb atenció la traça del teu codi per a un nombre de `discos` igual a `3`, intenta respondre a les següents qüestions sobre l'execució que del teu mètode realitza la JVM amb ajuda de la pila de crides.

Qüestions sobre l'activitat 3:

- Quina és la primera instrucció `moureDisc` que s'executa, la del cas base o la del general? Raona la teua resposta.
- Quants "post-it" és necessari apilar **abans d'**executar aquesta primera instrucció `moureDisc`? Coincideix amb el nombre de crides que s'han efectuat fins a eixe moment? Raona les teues respostes.
- Quants "post-it" és necessari apilar **després d'**haver executat la primera instrucció `moureDisc`?
- Quina és la segona instrucció `moureDisc` que s'executa, la del cas base o la del general? Raona la teua resposta i, després, contesta a les dos qüestions anteriors suposant que es plantegen per a este cas.
- Quina és l'última instrucció `moureDisc` que s'executa, la del cas base o la del general? Raona la teua resposta i, després, contesta a la segona i tercera qüestió anteriors suposant que es plantegen per a este cas.
- Basant-te en les respostes que has donat, podries demostrar que l'execució del mètode que has escrit acaba després d'haver-se realitzat un nombre de crides finit? Utilitza si vols en la teua demostració el concepte de funció limitadora, o si ho prefereixes intenta realitzar-la per inducció.

4.4 Activitat 4: Estimació del cost d'execució del mètode

Després d'haver comprovat que el teu mètode acaba la seua execució després d'un nombre de crides finit, esta quarta i última activitat et permetrà establir quant de temps tarda a fer-ho en funció del nombre de discos `N` per al que ho hages invocat: si un temps directament proporcional a `N`, o **lineal**, si un temps proporcional al quadrat de `N`, o **quadràtic**, si ... Per a això has de realitzar el `Pas 4` de la interfície, que il·lustra la següent figura, i apuntar el nombre de moviments de discos que es realitzen cada vegada que s'executa el teu mètode; a partir d'eixa sèrie de valors, hauràs d'induir la funció matemàtica a què pertanyen, la denominada Complexitat Temporal del mètode.



Serà llavors quan descobriràs per què el problema de les Torres d'Hanoi és conegut també com *el problema de la fi del món* i, a més, podràs resoldre sense problemes la qüestió següent:

> Conta la llegenda que en el temple de Brahma en Benarés existien tres agulles de diamants que servien per a apilar 64 discos d'or pur. Incansablement, els sacerdots transferien discos d'una agulla a una altra obeint sempre la llei de Brahma: *ningún disc es pot situar damunt d'un de menor diàmetre*.
>
> A l'inici del món els 64 discos es van disposar sobre la primera agulla, formant una torre de Brahma; segons la llegenda, en el moment en què el més xicotet dels discos es col·locara en la torre de destí, i es tinguera una nova torre de Brahma arribaria la fi del món.
>
> Segons la profecia dels brahmans, suposant que els monjos empraren 20 mseg. per a moure un disc, **quan arribarà la fi del món?**