Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

- Introducció als TADs
- Implementacions estàtiques i dinàmiques
- Emmagatzemament de TADs

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

La raó de ser...

Una col·lecció o contenidor és una abstracció que ens permet guardar un grup d'elements i fer les operacions necessàries per a gestionar-lo (afegir nous elements, consultar, eliminar, ...)

"Un TAD es pot definir com un conjunt de valors sobre els quals podem aplicar un grup d'operacions que compleixen determinades propietats."

TAD = Variables + Operacions (constructores i consultores)

Estructures de Dades

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

La raó de ser...

Aspectes a decidir en un TAD:

Quines dades es guarden	Com es guarden les dades
Quines són les operacions per a manipular les dades (i quines són les propietats que compleixen)	Com es programen les operacions
ESPECIFICACIÓ	**IMPLEMENTACIÓ**

Tipus **Abstracte** de Dades (TAD)

en el sentit **d'abstracció**, hem de poder treballar amb els valors del tipus, coneixent només les seves operacions i el seu funcionament (i desconeixent el codi que implementa l'operació)

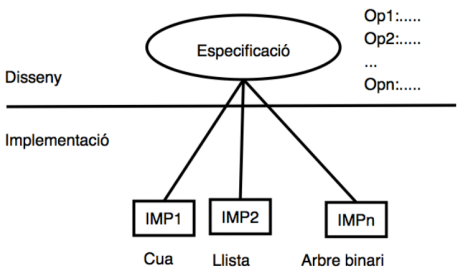
La manipulació dels objectes d'un tipus només ha de dependre del comportament descrit en l'especificació

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

1 Especificació ⟷ ***n Implementacions***

Quines dades es guarden	Com es guarden les dades
Quines són les operacions	Com es programen les operacions
L'especificació sempre és única	Però es pot implementar de moltes formes



La manipulació dels objectes d'un tipus només ha de dependre del comportament descrit en l'especificació ***i ha de ser independent de la seva implementació***

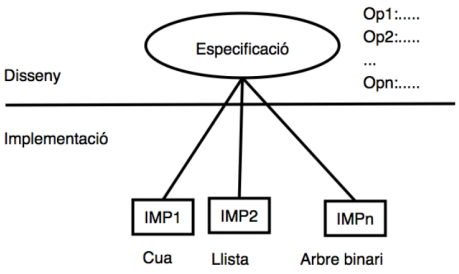
Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

*Com ha de ser l'**Especificació**?*

Quines dades es guarden
Quines són les operacions

- Precisa
- General
- Llegible
- No ambigua



Notació +/- formal – permetrà identificar sense ambigüitats el comportament de les operacions.

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

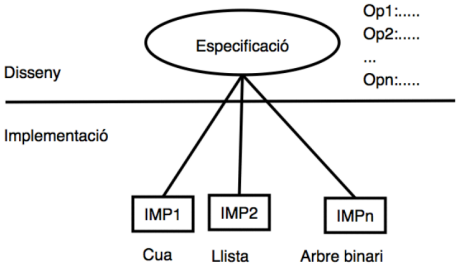
Tipus Abstractes de Dades

*Com ha de ser la **Implementació**?*

Com es guarden les dades

Com es programen les operacions

- Estructurada
- Eficient → Cost temporal, Cost espaial
- Llegible



La implementació és totalment transparent als usuaris del tipus.

La implementació ha de respectar l'especificació donada.

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

I perquè he de fer servir TADs...

Avantatges de treballar amb TADs

Abstracció: No hem de conèixer la implementació per a utilitzar el TAD
Correcció: Es poden avaluar els TADs per separat
Eficiència: Permeten escollir la implementació més adequada
Llegibilitat: Permeten entendre més ràpidament qualsevol codi
Reutilització: Es pot aprofitar un TAD en diferents codis
Modificabilitat: Més fàcils de manipular i depurar sense afectar a la resta de codi.
Organització: Permeten dividir la feina de programació en tasques independents.
Seguretat: Bloquegem l'accés directe a les dades, podem posar-hi controls.

Inconvenients de treballar amb TADs

Més feina a la part de disseny

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Relació entre el TAD i les Estructures de Dades

Una estructura de dades és la representació d'un TAD mitjançant la combinació dels tipus predefinits en els llenguatges i els constructors de tipus (vectors, ...)

Algunes combinacions són molt usades:

- Estructures lineals (seqüències)
- Multillistes (relacions binàries)
- Taules de dispersió (funcions i conjunts)
- Arbres (jerarquies)
- Grafs

La API de Java conté algunes d'aquestes estructures implementades de diferents formes. ***Hem d'aprendre a especificar-les i implementar-les?***

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Com abordem el treball amb un TAD?

Com fem l'especificació d'un TAD?
Com el podem implementar?
Criteris de decisió a l'hora d'escollir la implementació
Cost temporal
Cost espaial

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

ESPECIFICACIÓ d'un TAD

L'especificació és la definició de les operacions i el seu comportament

Per descriure una operació tenim la seva signatura

Signatura de l'operació

NOM: PARAMS_ENTRADA => PARAMS_SORTIDA

Per descriure el seu comportament:

Llenguatge natural
Especificacions pre/post: Lògica de predicats
Notació matemàtica: Especificació algebraica o equacional

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

ESPECIFICACIÓ d'un TAD

Descriure el comportament: Especificació algebraica o equacional

Defineix les propietats del TAD mitjançant equacions amb variables quantificades

Exemple: TAD conjunt d'enters

Operacions del TAD (signatura)

cBuit: => conjunt
afegir: conjunt, enter => conjunt
pertany: enter, conjunt => booleà
ple?: conjunt => booleà
#elems: conjunt => enter

Equacions

Descripció comportament de les operacions

Tipus d'operacions

Constructores
Consultores
Auxiliars/privades

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

cBuit: => conjunt
afegir: conjunt, enter => conjunt
pertany: enter, conjunt => booleà
ple?: conjunt => booleà
#elems: conjunt => enter

ESPECIFICACIÓ d'un TAD

Exemple: TAD conjunt d'enters

Descripció comportament de les operacions

Escollir un conjunt d'operacions com a constructores generadores (termes canònics)
Definir les equacions purificadores
Definir les equacions de la resta d'operacions
Definir les equacions d'error

Terme canònic

afegir(afegir(afegir(...(afegir(cbuit, e_1), e_2), ...), e_n), e_{n+1})
on $e_1 < e_2 < e_3 < ... < e_n < e_{n+1}$ => no hi ha repetits

Equacions purificadores

afegir(afegir(C, e), e) = afegir(C, e) *per evitar repetits*

[e≠e'] => afegir(afegir(C, e), e') = afegir(afegir(C, e'), e) *ordre d'inserció indiferent*

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

cBuit: => conjunt
afegir: conjunt, enter => conjunt
pertany: enter, conjunt => booleà
ple?: conjunt => booleà
#elems: conjunt => enter

ESPECIFICACIÓ d'un TAD

Exemple: TAD conjunt d'enters

Descripció comportament de les operacions

Equacions de la resta d'operacions

pertany(e, cBuit) = fals
pertany(e, afegir(C, e)) = cert
[e≠e'] => pertany(e, afegir(C, e')) = pertany(e, C)

ple?(C) = (#elems(C) = MAX_ELEMS)

#elems(cbuit) = Zero
[e ∈ C] => #elems(afegir(C, e)) = #elems(C)
[e ∉ C] => #elems(afegir(C, e)) = Succ(#elems(C))

Definir les equacions d'error

No(e∈ C) ∧ ple?(C) => afegir(C, e) = ERROR : Conjunt ple

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

ESPECIFICACIÓ d'un TAD

Descriure el comportament: combinació llenguatge natural i pre/post

Especificació

```
TAD ConjuntEnters {
    Crea un conjunt buit
    @pre cert
    @post El conjunt construït correspon al conjunt buit
    ConjuntEnters(int dimensio);

    Afegeix un element al conjunt. Si l'element ja hi és, no fa res
    @pre cert
    @post pertany(e)=cert (o element pertany al conjunt)
    @error si el conjunt està ple no es pot afegir l'enter
    void afegir(int e) throws conjuntPle;
```

Tota l'especificació al material de laboratori

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

IMPLEMENTACIÓ d'un TAD

Com organitzem la informació?
Quines operacions requereixen accés més ràpid?
Utilitzem memòria dinàmica o estàtica?
Implementem en estructures (structs de C) o en classes (Java)?
Com podem fer que el TAD sigui reutilitzable?

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Criteris de decisió a l'hora d'escollir la implementació

- Minimitzar el COST TEMPORAL: Que totes les operacions siguin molt ràpides. *No podem fer que totes les operacions, siguin instantànies, així que prioritzarem el cost de les operacions crítiques.*
- Minimitzar el COST ESPAIAL: Que gasti el mínim possible de memòria. *Avui en dia ja no hi ha les mateixes limitacions d'espai que abans.*

Per a calcular el cost espaial tindrem en compte les dades que hi ha introduïdes i la mida de l'estructura.

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Mesurar el COST TEMPORAL

Depèn de diferents factors:

- Màquina utilitzada
- Llenguatge de programació, compilador
- **Mida de les dades**

Podem fer el càlcul:

- Forma pràctica
- **Forma teòrica**

Funció de cost

$f: \mathbb{N} \rightarrow \mathbb{R}^+$

On N és la mida de les dades i R^+ el temps d'execució

No ens interessa el valor numèric del cost
Ens interessa **l'ordre de creixement** de la funció de cost
3n+5 o 5n+2 creixement ordre n, lineal

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Mesurar el COST TEMPORAL

Funció de cost

$f: \mathbb{N} \rightarrow \mathbb{R}^+$

L'objectiu és trobar una funció que a partir de la mida de les dades d'entrada, ens doni el cost temporal de l'algorisme.

- **Anàlisi del cas pitjor**
- Anàlisi del cas mitjà
- Anàlisi del cas millor

$O(n)$ Representa el conjunt de totes les funcions afitades superiorment per un múltiple positiu de f(n)

$\Theta(n)$ Representa el conjunt de totes les funcions que creixen exactament com f(n) (un múltiple positiu)

$2n + 7 \in O(n)$ $\quad 2n + 7 \in \Theta(n)$

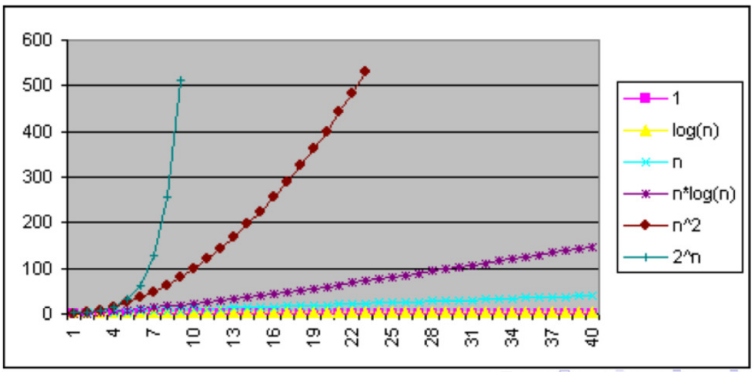
$2n + 7 \in O(n^2)$ $\quad 2n + 7 \notin \Theta(n^2)$

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Formes de creixement

$\Theta(1)$	Temps constant, independent mida dades	Escriure missatge
$\Theta(log(n))$	Temps logarítmic.	Cerca dicotòmica
$\Theta(n)$	Temps lineal	Cerca taula desordenada
$\Theta(n \cdot log(n))$	Temps quasi-lineal	Ordenar taula
$\Theta(n^k)$, k *const.*	Temps polinòmic (quadrat, cúbic,...)	Alguns algorismes de grafs
$\Theta(k^n)$, k *const.*	Temps exponencial 2^n, 3^n,...	Alguns problemes de satisfacció de restriccions.

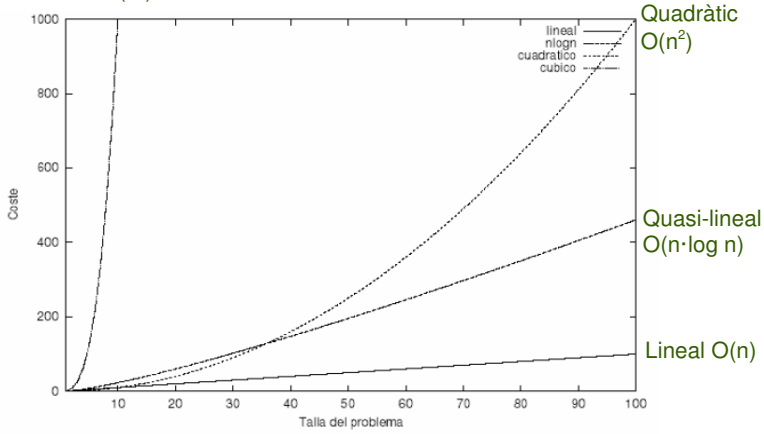


Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Formes de creixement

Cúbic O(n^3)



Quadràtic O(n^2)

Quasi-lineal O(n·log n)

Lineal O(n)

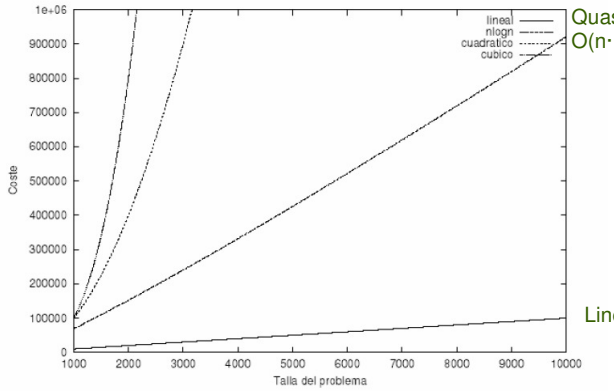
Quantitat de dades petita, per sota de 100 valors

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Formes de creixement

Cúbic $O(n^3)$ Quadràtic $O(n^2)$



Quasi-lineal $O(n \cdot \log n)$

Lineal $O(n)$

Quantitat de dades més gran

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Formes de creixement

n	log n	n	n log n	n^2	n^3	2^n	n!
10	1	10	10	100	1000	1024	3628800
100	2	100	200	10000	1000000	1E+30	9E+157
1.000	3	1.000	3.000	1000000	1E+9	1E+301	→∞
10.000	4	10.000	40.000	1E+8	1E+12	→∞	→∞
100.000	5	100.000	500.000	1E+10	1E+15	→∞	→∞

Relació entre el tamany del conjunt de dades (n) i el temps que es necessita per tractar-les segons la complexitat de l'algorisme

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Calcular el cost temporal d'un programa

- **Regla de la suma.**
  Si l'acció A_1 té el cost $t_1(n) \in \Theta(f_1)$ i l'acció A_2 té el cost $t_2(n) \in \Theta(f_2)$ el cost de la seqüència A_1, A_2 és $t_1(n) + t_2(n) \in \Theta(MAX(f1, f2))$.
- Exemples regla de la suma

$$n + 1 \in \Theta(n)$$
$$n + log(n) \in \Theta(n)$$
$$n + n = 2n \in \Theta(n)$$
$$n + n^2 \in \Theta(n^2)$$

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Calcular el cost temporal d'un programa

- **Regla del producte.**
  Si l'acció A_1 té el cost $t_1(n) \in \Theta(f_1)$ i l'acció A_2 té el cost $t_2(n) \in \Theta(f_2)$, aleshores $t_1(n) \cdot t_2(n) \in \Theta(f1 \cdot f2)$.
- Exemples regla del producte

$$n \cdot 1 \in \Theta(n)$$
$$n \cdot log(n) \in \Theta(n \cdot log(n))$$
$$n \cdot n = n^2 \in \Theta(n^2)$$
$$n \cdot n^2 = n^3 \in \Theta(n^3)$$

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Calcular el cost temporal d'un programa

Assignacions, comparacions, lectura o escriptura
Seqüències
Condicionals
Iteracions

Exemple: Número primer més gran no superior a …
Dissenyem un programa que donat un número N ens mostri per pantalla el numero el número primer P més gran que compleixi: P ≤ N

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Calcular el cost temporal d'un programa

```
public class PrimerMesGran {

public static void main(String[] args) {
  long valor=4275912996L;
  long tempsi, tempsf;
  tempsi=System.nanoTime();  // opc tempsi=System.currentTimeMillis();
  trobarPrimerMaxim(valor);
  tempsf=System.nanoTime();
  System.out.println("Per trobar el primer mes gran a "+valor+
              " ha tardat: "+(tempsf-tempsi)+" ns");
}
```

Hem de començar pels mètodes que es van cridant

Estructures de Dades

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

```
public static void trobarPrimerMaxim(long x) {
  if (x>3) {
   if (!esPrimer(x)) {
     if (x%2==0)
       x=x-1;
     else x=x-2;
     while (!esPrimer(x))
       x=x-2;
     System.out.println(x);
   } else System.out.println(x);
  } else System.out.println(x);
}
```


Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Θ(n)

```
public static boolean esPrimer(long aux) {
  boolean elDivideixen=false;
  if (aux%2==0)
    elDivideixen=true;
  else {
    long fin=(long)Math.sqrt(aux); fin=fin+1;
    long seguent=3;
Θ(n)  while ((seguent<fin)&& !elDivideixen) {
      if (aux%seguent==0)          Θ(1)
        elDivideixen=true;         Θ(1)
        seguent=seguent+2;         Θ(1)
    }
  }
  return(!elDivideixen);
}
```

La iteració es repeteix n vegades
Regla del producte: Θ(n*1)=Θ(n)

Regla de la suma:
màxim(Θ(1), Θ(1)) = Θ(1)

Estructures de Dades

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

$\Theta(n^2)$

```
public static void trobarPrimerMaxim(long x) {
  if (x>3) {                    Θ(1)
   if (!esPrimer(x)) {          Θ(n)
     if (x%2==0)
       x=x-1;
     else x=x-2;
     while (!esPrimer(x))       Θ(n)
       x=x-2;                   Θ(1)
     System.out.println(x);
   } else System.out.println(x);
  } else System.out.println(x);
}
```

La iteració es repeteix n vegades
Regla del producte: $\Theta(n*n)=\Theta(n^2)$

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Calcular el cost temporal d'un programa

$\Theta(n^2)$

```
public class PrimerMesGran {

public static void main(String[] args) {
  long valor=4275912996L;
  long tempsi, tempsf;
  tempsi=System.nanoTime();  // opc tempsi=System.currentTimeMillis();
  trobarPrimerMaxim(valor);   Θ(n²)
  tempsf=System.nanoTime();
  System.out.println("Per trobar el primer mes gran a "+valor+
              " ha tardat: "+(tempsf-tempsi)+" ns");
}
```

```
<terminated> PrimerMesGran [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (27 set. 2016 , 17:50:35)
4275912661
Per trobar el primer mes gran a 4275912996 ha tardat: 12459154 ns
```

Escola Tècnica Superior d'Enginyeria
Universitat Rovira i Virgili

Tipus Abstractes de Dades

Podríem tenir un algoritme de calcular el primer menys optimitzat. A nivell de cost seria del mateix ordre però el temps d'execució canviaria una mica.

Θ(n)

```
/* força bruta*/
public static boolean esPrimer(long aux) {
  boolean elDivideixen=false;
  long seguent=2;
  while ((seguent<aux)&& !elDivideixen) {
    if (aux%seguent==0)
      elDivideixen=true;
    seguent=seguent+1;
  }
  return(!elDivideixen);
}
```

```
<terminated> PrimerMesGran [Java Application] C:\Program Files\Java\jre8\bin\javaw.exe (27 set. 2016 , 18:16:54)
4275912661
Per trobar el primer mes gran a 4275912996 ha tardat: 128475978964 ns
```