Diseño de algoritmos

UNIVERSIDAD DE LOS ANDES

Diseño y Análisis de Algoritmos

Cátedra de Programación

Carrera de Ingeniería de Sistemas

Prof. Isabel Besembel Carrera

Diseño de algoritmos

- Es una **actividad creativa** que no tiene una receta exitosa y precisa.
- Existen muchos problemas importantes para los que no se conoce un algoritmo eficiente
- Clasificando los algoritmos según patrones estructurales similares, es posible identificar ciertas estrategias que normalmente arrojan **algoritmos eficientes y correctos**

1. Proceso de diseño

Estrategias:

1. Estar familiarizado con un repertorio de problemas estándares

❖ Producir una especificación clara del problema

Ignorando detalles como formatos, se debe centrar la atención en lo que parece ser la parte esencial del problema

❖ Expresar el problema en una forma abstracta

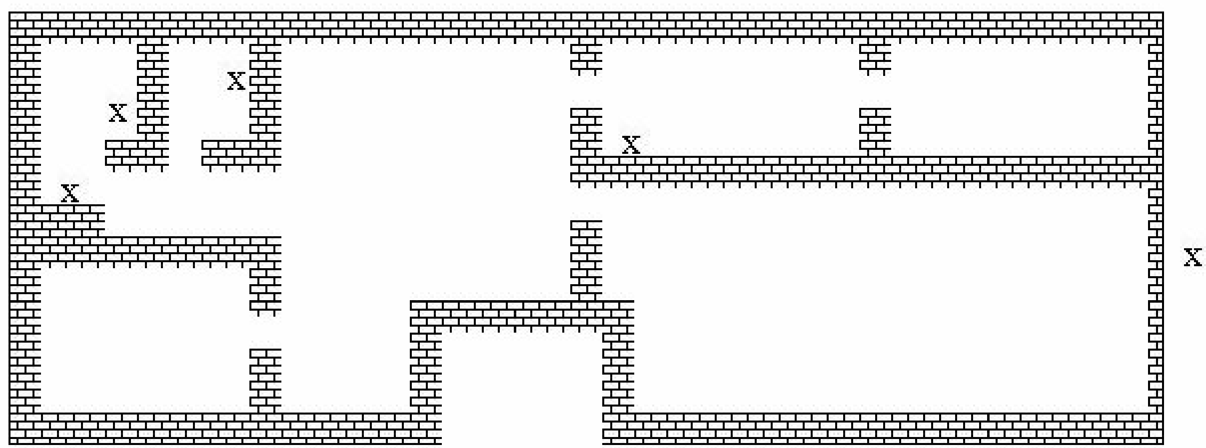
Normalmente se obtiene uno estándar y no es necesaria innovación alguna

Estrategia 1

Ejemplo:

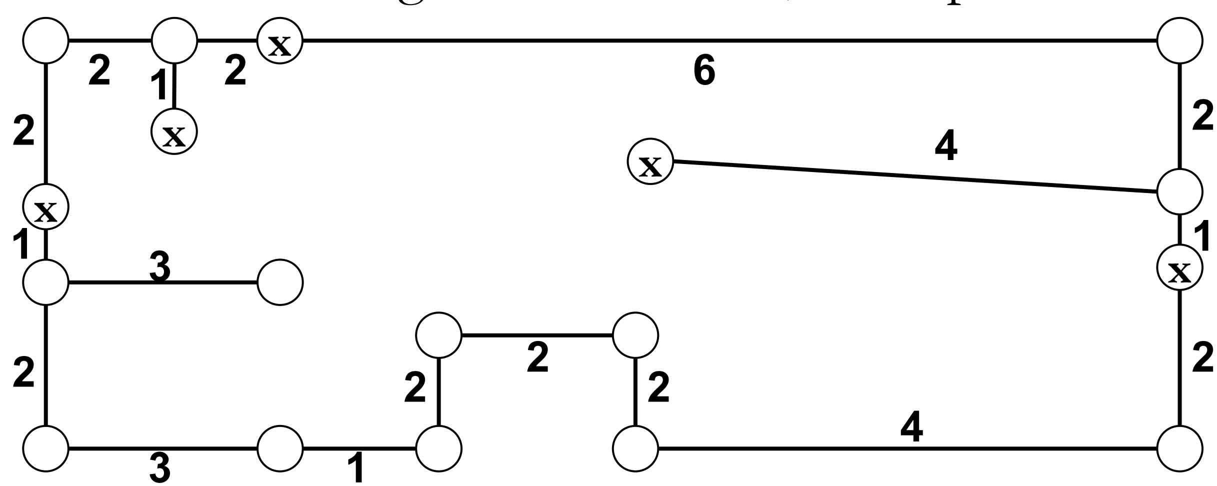
un arquitecto produce el plano mostrado y se desea conocer ¿cuál es la forma más económica de colocar las tuberías de agua?

Los lugares donde se necesita agua y por donde llega a la casa están marcados con una **x**, y solo se pueden colocar tubos en las paredes



Estrategia 1

- Después de abstraer el problema se tiene:
  - Dado un grafo G con costos en sus arcos, encontrar la forma de conectar un subconjunto de sus nodos tal que minimice el costo total de los arcos utilizados
- Problema estándar: Encontrar el árbol de Steiner para G
- No se le conoce aún un algoritmo eficiente, es un problema NP



Estrategia 2

2. Utilizar algoritmos incrementales
   - Una instancia de un problema es como un territorio desconocido
   - La vía más natural es examinar alguna parte del problema, luego tomar la próxima parte y repetir esto hasta que no hayan partes que examinar.
   - Esta estrategia incremental se expresa como:

Junio,04		
	incremental(Tipo: instancia):tipoDeResultado **{ pre: precondiciones }** **{ pos: poscondiciones }**	
1 2 3	**resultado=valor inicial** **(instancia≠vacía)[X=un elemento de la instancia del problema** **eliminar X de la instancia del problema** **actualizar resultado para que refleje la solución X]** **regrese resultado**	**Documentación**

UNIVERSIDAD
DE LOS ANDES

Estrategia 2

- Cuando el diseñador/programador decide utilizar esta estrategia debe a continuación considerar **cómo escoger X en cada paso**
  - **Algoritmos incrementales de tipo 1**

    Es irrelevante la escogencia de X

    - **Ventaja**: la selección de X es simple y eficiente
    - **Desventaja**: el algoritmo es ciego, pues no conoce nada de los valores del elemento X que ha seleccionado

    Se caracterizan por tener como **invariante del lazo**:

    - ***resultado*** es una solución completa del subproblema, representado por la parte de la instancia que ha sido eliminada

UNIVERSIDAD
DE LOS ANDES

Estrategia 2, tipo 1

- Ejemplos:
  1. Encontrar la suma de los elementos de un arreglo

Junio, 2004

suma(Arreglo(N)DeNatural v, Natural a, Natural b):Natural

{pre: $a \leq b+1$ } {pos: $s \geq 0$ }

s = 0 (s = s + v(i)) i = a, b, 1 regrese s	➢**i**: Natural. **Contador**. ➢**s:** Natural. Acumulador que tendrá la suma de números en el arreglo v.

1	v = (4, 6, 8, 23), a = 1, b = 4 -> s = 41
2	v = ( ), a = 0, b = 0 -> s = 0

Invariante del lazo: "s contiene el total de todos los elementos examinados"

Estrategia 2, tipo 1

	v
a=1	5
	23
	6
	28
	45
	36
	9
	5
	47
	8
b=11	22

inicio

i = ?	s = 0
i = 1	s = 5
i = 2	s = 28
i = 3	s = 34
i = 4	s = 62
i = 5	s = 107
i = 6	s = 143
i = 7	s = 152
i = 8	s = 157
i = 9	s = 204
i = 10	s = 212
i = 11	s = 234

a=4

b=6

i = ?	s = 0
i = 4	s = 28
i = 5	s = 73
i = 6	s = 109

Invariante del lazo: “s contiene el total de todos los elementos examinados”

instancia={9,2,6}
1era iteración resultado={9}
2da iteración resultado={2,9}
resultado={2, 6, 9}

…tegia 2, tipo 1

2. Ordenar en forma ascendente un conjunto de números con el método de inserción simple

Junio, 2004 insercionSimple(Arreglo(N)DeNatural v, Natural a, Natural b) {pre: $a \leq b+1$ } {pos: $s \geq 0$ }		
1	((si (v(j) < v(j-1)) entonces k, v(j), v(j-1) = v(j), v(j-1), k fsi) j = i+1, a, -1) i = a, b, 1	➢**i, j**: Natural. **Contadores**. ➢**k:** Natural. Variable auxiliar para realizar el intercambio.
1 2	v = (4, 6, 8, 23), a = 1, b = 4 -> v = (4,6,8,23) v = ( ), a = 0, b = 0 -> v = ( )	Caso exitoso Arreglo vacío

Estrategia de ordenamiento: como ordenar las cartas en una mano cuando las reparten, cada nueva carta se coloca en el sitio que le corresponde

UNIVERSIDAD DE LOS ANDES

Estrategia 2, tipo 1

resultado {5, 23}

	V	V	V	V	V	V	V	V	V	V	V
a=1	5	5	5	5	5	5	5	5	5	5	5
	23	23	6	6	6	6	6	5	5	5	5
	6		23	23	23	23	9	6	6	6	6
	28			28	28	28	23	9	9	8	8
	45				45	36	28	23	23	9	9
	36					45	36	28	28	23	22
	9						45	36	36	28	23
	5							45	45	36	28
	47								47	45	36
	8									47	45
b=11	22										47
	inicio	i=1 j=2	i=2 j=3	i=3 j=4	i=4 j=5	i=5 j=6	i=6 j=7	i=7 j=8	i=8 j=9	i=9 j=10	i=10 j=11

Estrategia 2, tipo 2

- **Algoritmos incrementales de tipo 2**

  La escogencia de X es primordial para evitar la reactualización de lo hecho anteriormente.

  Invariante del lazo que los caracteriza:

  - ***resultado*** es una parte de la solución global de la instancia del problema, que podrá ser añadida pero no modificada

- Ejemplos:

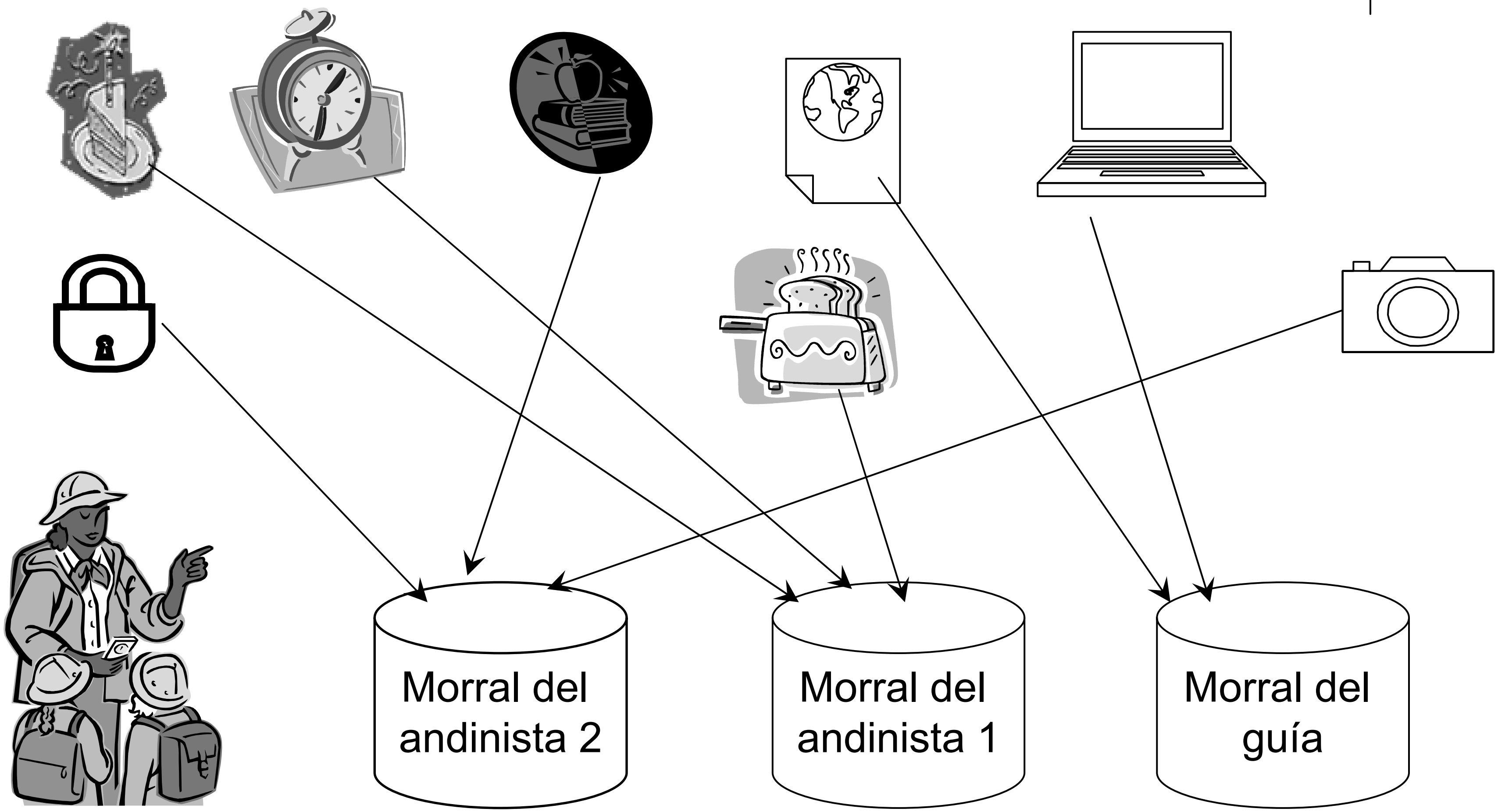
1. Equilibrar el peso de los morrales de un grupo de Andinistas

Estrategia 2, tipo 2

- Se tiene un número de items de pesos variados y se desea que todos los morrales tengan, en lo posible, el mismo peso
- Suponga que ya hay varios items empacados y los morrales tienen aproximadamente el mismo peso, no es simple escoger el próximo item!!! Si éste es muy pesado, reempacar es inevitable!!!
- **Estrategia**: seleccionar los items más pesados primero y tratar de equilibrar con los livianos

UNIVERSIDAD DE LOS ANDES

Estrategia 2, tipo 2



instancia={9,2,6}
1era iteración resultado={2,9,6}
2da iteración resultado={2,6,9}

Estrategia 2, tipo 2

2. Ordenar en forma ascendente un conjunto de números con el método de selección simple

Junio, 2004 seleccionSimple(Arreglo(N)DeNatural v, Natural a, Natural b) {pre: $a \leq b+1$ } {pos: $s \geq 0$ }		
1	((si (v(j) < v(i)) entonces min, pmin = v(j), j fsi) j = i+1, b-1, 1 v(pmin), v(i) = v(i), min) i = a, b, 1	➤**i, j**: Natural. **Contadores**. ➤**min:** Natural. Variable auxiliar para realizar el intercambio. ➤**pmin:** Natural. Posición del valor mínimo
1 2	v=(4, 6, 8, 23), a=1, b=4 -> v=(4,6,8,23) v = ( ), a = 0, b = 0 -> v = ( )	Caso exitoso Arreglo vacío

Estrategia de ordenamiento: se selecciona el más pequeño y se coloca de primero, luego el siguiente más pequeño y se coloca de segundo, ….

Estrategia 2, tipo 2

	v	v	v	v	v	v	v	v	v	v	v
a=1	5	5	5	5	5	5	5	5	5	5	5
	23	23	23	5	5	5	5	5	5	5	5
	6	6	6	6	6	6	6	6	6	6	6
	28	28	28	28	28	8	8	8	8	8	8
	45	45	45	45	45	45	9	9	9	9	9
	36	36	36	36	36	36	36	22	22	22	22
	9	9	9	9	9	min 9	45	45	23	23	23
	5	5	min 5	23	23	23	23	min 23	45	28	28
	47	47	47	47	47	47	47	47	47	47	36
	8	8	8	8	min 8	28	28	28	min 28	45	45
b=11	22	22	22	22	22	22	min 22	36	36	min 36	47
	inicio	i=1 j=2	i=2 j=3	i=3 j=4	i=4 j=5	i=5 j=6	i=6 j=7	i=7 j=8	i=8 j=9	i=9 j=10	i=10 j=11

Ejercicios

1. Dado un arreglo bidimensional p(nxm)[a…b, c…d] donde se sabe que $p(i, j) \leq p(i, j+1)$ y $p(i, j) \leq p(i+1, j)$ $\forall$ i, j. Lo que significa que las entradas estan ordenadas a lo largo de las filas y columnas. Diseñe un algoritmo eficiente para determinar si el valor x está en p.
2. Un centro de computación tienen una única impresora. En cierto momento, n usuarios envían sus archivos a imprimir y esperan la salida impresa. La impresora debe decidir su plan de impresión de manera de minimizar el tiempo total de espera de los n usuarios. El tamaño de cada archivo es conocido y con ello se puede calcular t(x): tiempo que toma imprimir el archivo x. Se propone el algoritmo siguiente que imprime los archivos por orden de tamaño. Pruebe que este algoritmo produce el plan óptimo.

Junio, 2004 planOptimo() {pre:}		{pos:}
1 2	Y = {todos los archivos} (Y ≠ {}) [remover de Y el archivo x de min t(x) imprimir (x)]	➢**Y**: Conjunto de archivos. Archivos a imprimir ➢**x:** Archivo. Archivo en Y ➢**t(x):** real. Tiempo de impresión de x

Ejercicios

3. Diseñar un algoritmo para imprimir todos los subconjuntos de un conjunto finito dado. Ejemplo: dado el conjunto {a, b, c} todos los subconjuntos son

   {} {a} {b} {c}

   {a, b} {a, c} {b, c}

   {a, b, c}