Ordenamiento en 1D y 2D

Note Title 4/6/2010

I Algoritmos de ordenamiento

	Peor$_n$	Mejor$_n$	Adaptativo?	
MergeSort	$n \lg n$	$n \lg n$	$O(n \lg n)$	
AMergeSort	$n \lg n$	n	$O(n \lg p)$	p fijo
HeapSort	$n \lg n$	$n \lg n$	$O(n \lg n)$	
SplayTreeSort	$n \lg n$	n	( )	
Quicksort	n^2	$n \lg n$		
StrongInsertionSort	n^2	n	$Inv = \{(i,j) \text{ tq } i<j,\ A[i]>A[j]\}$	
Local InsertionSort	$n \lg n$	n	$n \lg(Inv/n)$	

II Problema de Ordenamiento

Comparar Medidas

algorítmicamente más fino

① (P_b, M_1) es algorithmically finer que (P_b, M_2) "$\leq$"

$(P_b, M_1) \leq_{alg} (P_b, M_2)$ si y solamente si

$\forall$ algoritmo M_1 optimal es M_2-optimal.

② (P_b, M_1) y M_2 son equivalente si

$(P_b, M_1 \leq_{Alg} (P_b, M_2)$ y $(P_b, M_2 \leq_{Alg} (P_b, M_1)$

Optimalidad

Un algoritmo A es (P_b, η_1) optimal si no hay un algoritmo B para P_b con mejor Peor caso por η_1 fijo.

Ejemplo DiskPile : $\eta_1 = h, n_1, \dots, n_h$
o $\eta_1 = h, n$

Como mostrar la optimalidad?

① Cota inferior por (P_b, Π_n) fija sobre la cantidad de elementos distintos / instancias

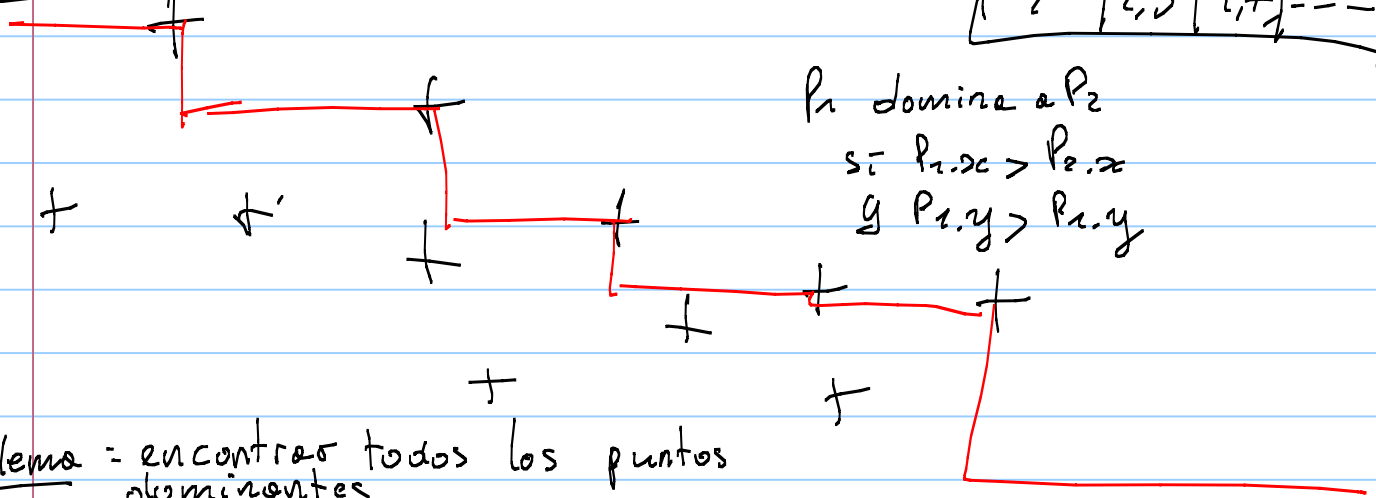
$\Rightarrow$ cota inferior sobre la complejidad en el modelo de comparacion.

② Analisis de Adversarie $\rightarrow$ Optimalidad ("Instance Optimality") al nivel de las Instancias

II. Ordenamiento en 2 Dimensiones

① Max

A [1,5 | 2,3 | 4,7 ---



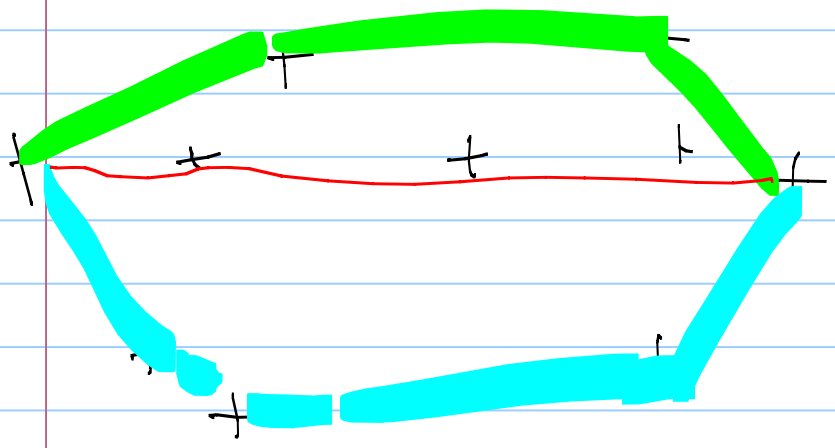
P_1 domina a P_2
si $P_1.x > P_2.x$
y $P_1.y > P_1.y$

Problema = encontrar todos los puntos dominantes.

Algoritmo (entre otros)

1. Ordenar $O(n \lg n)$!
2. Scan $O(n)$

② Convex Hull / Upper Hull 2D



① Ordenar $O(n \lg n)$?

② Scan. $O(n)$