Minimo Maximo de un arreglo

Jeremy Barbay

11 March 2011

Contents

1 Preliminary Readings

- Cotas Inferiores en apuntes de CC3001
  - http://www.dcc.uchile.cl/~bebustos/apuntes/cc3001/Ordenacion/#1

2 Cota inferior

- Sean las variables siguentes:
  - O los o elementos todavia no comparados;
  - G los g elementos que "ganaron" todas su comparaciones hasta ahora;
  - P los p elementos que "perdieron" todas su comparaciones hasta ahora;
  - E las e valores eliminadas (que perdieron al menos una comparacion, y ganaron al menos una comparacion);
- (o, g, p, e) describe el estado de cualquier algoritmo:

  - siempre $o+g+p+e=n$;
  - al inicio, $g=p=e=0$ y $o=n$;
  - al final, $o=0$, $g=p=1$, y $e=n-2$.

- Despues una comparacion $a?b$ en cualquier algoritmo del modelo de comparacion, (o,g,p,e) cambia en funcion del resultado de la comparacion de la manera siguente:

	$a\in O$	$a\in G$	$a\in P$	$a\in E$
$b\in O$	$o-2,g+1,p+1,e$	$o-1,p,e+1$ $o-1,g,p+1,e$	$o-1,g,p,e+1$ $o-1,g+1,p,e$	$o-1,g+1,p,e$ $o-1,g,p+1,e$
$b\in G$		$o,g-1,p,e+1$	o,g,p,e $o,g-1,p-1,e+2$	o,g,p,e $o,g-1,p,e+1$
$b\in P$			$o,g,p-1,e+1$	o,g,p,e $o,g,p-1,e+1$
$b\in E$				o,g,p,e

- En algunas configuraciones, el cambio del vector estado depende del resultado de la comparacion: un adversario puede maximisar la complejidad del algoritmo eligando el resultado de cada comparacion. El arreglo siguente contiene en graso las opciones que maximizan la complejidad del algoritmo:

	$a\in O$	$a\in G$	$a\in P$	$a\in E$
$b\in O$	$o-2,g+1,p+1,e$	$o-1,p,e+1$ $\mathbf{o-1,g,p+1,e}$	$o-1,g,p,e+1$ $\mathbf{o-1,g+1,p,e}$	$o-1,g+1,p,e$ $o-1,g,p+1,e$
$b\in G$		$o,g-1,p,e+1$	$\mathbf{o,g,p,e}$ $o,g-1,p-1,e+2$	$\mathbf{o,g,p,e}$ $o,g-1,p,e+1$
$b\in P$			$o,g,p-1,e+1$	$\mathbf{o,g,p,e}$ $o,g,p-1,e+1$
$b\in E$				o,g,p,e

- Con estas opciones, hay
  - $\lceil n/2\rceil$ transiciones de O a $G\cup P$, y
  - $n-2$ transiciones de $G\cup P$ a E.

- Eso resulta en una complejidad en el peor case de $\lceil 3n/2\rceil-2\in 3n/2+O(1)$ comparaciones.

3 Cota superior

- Calcular el minimo con el algoritmo previo, y el maximo con un algoritmo simetrico, da una complejidad de $2n-2$ comparaciones, que es demasiado.
- El algoritmo siguente calcula el max y el min en $\frac{3n}{2}-2$ comparaciones:
  1. Dividir A en $\lfloor n/2 \rfloor$ pares (y eventualemente un elemento mas, x).
  2. Comparar los dos elementos de cada par.
  3. Ponga los elementos superiores en el grupo S, y los elementos inferiores en el grupo I.
  4. Calcula el minima m del grupo I con el algorimo de la pregunta previa, que performa $\lfloor n/2 \rfloor - 1$ comparaciones
  5. Calcula el maxima M del grupo I con un algoritmo simetrico, con la misma complejidad.
  6. Si n es par,
     - m y M son respectivamente el minimo y el maximo de A.
  7. Sino, si $x < m$,
     - x y M son respectivamente el minimo y el maximo de A.
  8. Sino, si $x > M$,
     - m y x son respectivamente el minimo y el maximo de A.
  9. Sino
     - m y M son respectivamente el minimo y el maximo de A.
- La complejidad total del algoritmo es
  - $n/2 + 2(n/2-1) = 3n/2 - 2 \in 3n/2 + O(1)$ si n es par
  - $(n-1)/2 + 2(n-1)/2 + 2 = 3n/2 + 1/2 \in 3n/2 + O(1)$ si n es impar.
  - en la clase $3n/2 + O(1)$ en ambos casos.