

How to form recurrence relations (RRs) for execution time of algorithms?

Selection Sort:

①

In iteration i identify max in $A[i \dots n-i+1]$ and swap it with $A[n-i+1]$.

① \rightarrow $n-1$ comparisons when $i=1$, i.e. the size of the array = n ($A[1 \dots n]$).

We repeat this for all ' i '.

Thus $T(n) = T(n-1) + n-1$

Time for input of size n

Time for input of size $n-1$

$(n-1)$ amount of work is done to reduce problem size to $n-1$ from n .

Solving gives $T(n) = n(n-1)/2$
 $= \Theta(n^2)$
 $= O(n^2)$
 $= SL(n^2)$.

Mergesort

$$MS(i, j) \begin{cases} \text{if } (i=j) \text{ return } i; \\ MS(i, (i+j)/2); \\ MS((i+j)/2 + 1, j); \end{cases}$$

Requires $\Theta(n)$ \leftarrow Merge $(i, \frac{i+j}{2}, \frac{i+j}{2} + 1, j)$;

$$T(n) = 2T(n/2) + an \rightarrow \text{corr to } \Theta(n).$$

Where n is size i.e. $j-i+1$

$$A[i, (i+j)/2] \text{ has } \frac{i+j}{2} - i + 1 = \frac{j-i}{2} + 1$$

Like wise $A[(i+j)/2 + 1, j]$ has $\sim n/2$ elements.

We ignore ± 1 .