

1953729 梁浩泽

1. (a) $T(N) = 2T(\frac{N}{2}) + N \log N$ $T(1) = 1$

存在 $\varepsilon = 1$ 且 $T(N) = \Theta(N \log N)$ 则有 $T(n) = \Theta(n \log_2 n)$

(b) $T(N) = 4T(\frac{N}{2}) + N$ $T(1) = 0$

存在 $\varepsilon = 1$ 且 $T(N) = O(N)$ 则有 $T(n) = \Theta(n^2)$

(c) $T(N) = T(\frac{N}{2}) + 2^N$ $T(1) = 1$

存在 $\varepsilon > 0$ 且 $T(n) = \Omega(n^\varepsilon)$ 同时存在 $c < 1$ 使 $f(\frac{n}{2}) \leq cf(n)$

则有 $T(n) = \Theta(2^n)$

2. 合并排序

mergesort(A[0...n-1])

if $n > 1$

copy A[0...[n/2]-1] to B[0...[n/2]-1]

copy A[n/2...n-1] to C[0...[n/2]-1]

mergesort(B[0...[n/2]-1])

mergesort(C[0...[n/2]-1])

merge(B, C, A)

时间复杂度 $T(n) = O(n \log n)$

空间复杂度 $S(n) = O(n)$

merge(B[0...p-1], C[0...q-1], A[k...p+q-1])

$i \leftarrow 0; j \leftarrow 0; k \leftarrow 0$

while $i < p$ and $j < q$ do

if $B[i] \leq C[j]$

$A[k] \leftarrow B[i]; i \leftarrow i + 1$

else $A[k] \leftarrow C[j]; j \leftarrow j + 1$

$k \leftarrow k + 1$

if $i = p$

copy C[j...q-1] to A[k...p+q-1]

else copy B[i...p-1] to A[k...p+q-1]

3.

MaxKnapsack(n, c, w[], p[]){

$cw = 0$ $cp = 0$

bestp = 0

$i = 1$ $up = \text{Bound}(1)$

while ($i \neq n+1$)

if ($cw + w[i] \leq c$)

if ($cp + p[i] > \text{bestp}$)

$\text{bestp} = cp + p[i];$

AddLiveNode($up, cp + p[i] + cw + w[i], \text{true}, i+1$);

$up = \text{Bound}(i+1);$

if ($up \geq \text{bestp}$)

AddLiveNode($up, up, cw, \text{false}, i+1$);

$H \rightarrow \text{DeleteMax}(N);$

$i = N.\text{Level};$

Bound(i){

$\text{cleft} = c - cw$

$b = cp$

while ($i \leq n$ && $w[i] \leq \text{cleft}$) {

$\text{cleft} -= w[i];$

$b += p[i];$

$i++;$ }

if ($i \leq n$)

$b += p[i] / w[i] * \text{cleft};$

return b; }

