2019320139 최윤지

#1-1

1) Creat an ADT, Set

ADT *Set* is

objects: a group with zero or more elements that meet certain conditions

functions:

for all *set, set1, set2* ∈ *Set, item* ∈ *element*

Set Create() ::==

create an empty set and **return**

Set Insert(*set, item*) ::==

if(IsIn(*set, item*)) **return** *set*

else insert *item* into *set* and **return** *set*

Set Remove(*set, item*) ::==

if(IsIn(set, item)) remove *item* from *set* and **return** *set*

else return *set*

Boolean IsIn(*set, item*) ::==

if(*item* ∈ *set*) *return* TRUE

else *return* FALSE

Set Union(*set1, set2*) ::==

create an set with elements that either *set1* or *set2* have

Set Intersection(*set1, set2*) ::==

create an set with elements that both *set1* and *set2* have

Set Difference(*set1, set2*) ::==

if(*set1* is empty set) return empty set

else create an set with elements that *set1* have but *set2* doesn't have

2) Create an ADT, Bag

ADT *Bag* is

objects: a group with zero or more elements that meet certain conditions and be able to be duplicate

functions:

for all *bag* ∈ *Bag, item* ∈ *element*

Bag Create() ::==

create an empty bag and **return**

Bag Insert(*bag, item*) ::==

insert item into *bag* and **return**

Bag Remove*(bag, item*) ::==

if(IsIn(*bag, item*)) remove *item* from *bag* and **return** *bag*

else return *bag*

Boolean IsIn(*bag, item*) ::==

if(*item* $\in$ *bag*) **return** TRUE

else return FALSE

#1-2

step count table

Statement	s/e	Frequency	Total steps
`void multi(int a[][MAX_SIZE], int b[][MAX_SIZE], int c[][MAX_SIZE])`	0	0	0
`{`	0	0	0
`int i, j, k;`	0	0	0
`for(i=0; i<MAX_SIZE;i++){`	1	MAX_SIZE+1	MAX_SIZE+1
`for(j=0; j<MAX_SIZE;j++){`	1	MAX_SIZE×(MAX_SIZE+1)	MAX_SIZE×(MAX_SIZE+1)
`c[i][j] = 0;`	1	MAX_SIZE×MAX_SIZE	MAX_SIZE×MAX_SIZE
`for(k=0; k<MAX_SIZE; k++){`	1	MAX_SIZE×MAX_SIZE×(MAX_SIZE+1)	MAX_SIZE×MAX_SIZE×(MAX_SIZE+1)
`c[i][j] += a[i][k]*b[k][j];`	1	MAX_SIZE×MAX_SIZE×MAX_SIZE	MAX_SIZE×MAX_SIZE×MAX_SIZE
`}`	0	0	0
`}`	0	0	0
`}`	0	0	0
`}`	0	0	0
Total	2(MAX_SIZE)³ + 3(MAX_SIZE)² + 2MAX_SIZE + 1		

#1-3

Show that the following statements are incorrect

1) $10n^2+9=O(n)$

For every positive c, $10n^2+9 \geqq cn$ when $n \geqq \frac{c}{10}$.

So, there don't exist c and n_0 that satisfy $10n^2+9 \leqq cn$ for all n, $n \geqq n_0$.

$\therefore 10n^2+9 \neq O(n)$

2) $n^2\log n = \Theta(n^2)$

For every positive c, $n^2\log n \geqq cn^2$ when $n \geqq 2^c$

So, there don't exist c_2 and n_0 that satisfy $c_1n^2 \leqq n^2\log n \leqq c_2n^2$ for all n,

$n \geqq n_0$.

$\therefore\ n^2 \log n \neq \Theta(n^2)$

3) $n^2/\log n = \Theta(n^2)$

For every positive c, $n^2/\log n \leqq cn^2$ when $n \geqq 2^{\frac{1}{c}}$.

So, there don't exist c_1 and n_0 that satisfy $c_1 n^2 \leqq n^2/\log n \leqq c_2 n^2$ for all n, $n \geqq n_0$.

$\therefore n^2/\log n \neq \Theta(n^2)$

4) $n^3 2^n + 6n^2 3^n = O(n^2 2^n)$

For every positive c, $n^3 2^n + 6n^2 3^n \geqq cn^2 2^n$ when $n \geqq c - 6$

So, there don't exist c and n_0 that satisfy $n^3 2^n + 6n^2 3^n \leqq cn^2 2^n$ for all n, $n \geqq n_0$

$\therefore n^3 2^n + 6n^2 3^n \neq O(n^2 2^n)$

5) $3^n = O(2^n)$

For every positive c, $3^n \geqq c \bullet 2^n$ when $n \geqq \log_{\frac{3}{2}} c$

So, there don't exist c and n_0 that satisfy $3^n \leqq c \bullet 2^n$ for all n, $n \geqq n_0$.

$\therefore\ 3^n \neq O(2^n)$