2014121136

구한모 Hanmo Ku

1.

a) if $f(n) \in O(g(n))$ and $g(n) \in O(f(n))$, then $f(n) <= cg(n)$ and $g(n) <= c'f(n)$. $g(n) <=$ is equivalent to that $1/c' * g(n) <= f(n)$. Therefore $1/c' * g(n) <= f(n) <= c\, g(n)$ and $f(n) \in \Theta(g(n))$

b) If $f(n) = 5n^2 + 3n + 2$ then $f(n) \in O(n^2)$ and $f(n) \in \Omega(n^2)$. If $g(n) = 4n^2+12n+8n^{(1/2)}$ then $g(n) \in O(n^2)$ and $g(n) \in \Omega(n^2)$. Therefore c1, c2 exists such that $g(n) <= c2n^2$ and $c1n^{}$ $<= g(n)$. This is equivalent to that $1/c1\, g(n) <= n^2$ and $n^2 <= 1/c2\, g(n)$. Also, c3, c4 exists such that $f(n) >= c3n^2$ and $f(n) <= c4n^2$. Therefore $c3/c1\, g(n) <= f(n)$ and $c4/c2\, g(n) >= f(n)$. Therefore $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$.

2.

Let's assume that the size of the list of objects L is n. At the worst case, the comparing operation happens for $(1+2+\ldots+(n-2)+(n-1))$times, which is $n*(n-1)/2$ times. So the complexity is $O((n^2)*(k^2))$

3.

Algorithm : Recursive Multiply

Input : integer x, y

Ouput : multiplication of x and y

if y = 1 **then** return x;

else if y = -1 **then** return -x;

else if y>1 **then** return Recursive Multiply (x, y-1) + x;

else if y<-1 **then** return Recursive Multiply (-x, -y);

4.

a)

Algorithm : Split List

Input : linked list L, index i

Output : linked list X and Y

Lsize 🡸 the size of L;

head of new list X 🡸 L(i+1);

tail of X 🡸 tail of L;

the size of X 🡸 Lsize – i -1;

tail of L 🡸 L(i);

the next node of L(i) 🡸 null;

the size of L 🡸 i+1;

new linked list y🡸L;

return X and Y;

the running time is O(n) (At the worst case, searching for L(i+1) operates for n times)

b) O(n)

5.

Algorithm : Perfect Tree dertermination

Input : Tree T and integer k

Output : the tree level p of tree T which makes T as perfect Tree at the level p by removing some nodes.

```
If number of the children of the root of tree T < k
        Return 1;
Else
        min ← the depth of the tree T
        for i←0 to k-1
                root of new Tree Tt ← the i-indexed child of the root of the tree T;
                depth of tree Tt ← depth of T -1;
                Temp ← call Perfect Tree determination(Tt, k);
                        If temp<min, min←temp;
        return min+1
```

6

a)

Assume that the relation given is a total order relation, then the relation is valid for any real number x1, x2, y1, y2, z1, z2 where point a = (x1, y1, z1) and b = (x2, y2, z2). Let a be a = (1, 0, 0)and b be b= (0, 0, 1). Since a and b belong to the total order relation and a != b, |a| must not be equal to |b|. However |a| = (1+0+0)^(1/2) = 1 and |b| = (0+1+0)^(1/2) = 1, so |a| =|b| therefore, the relation given in the problem is not a total order relation.

b)