

a) $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1$ là $O(\log n)$ (NOTE : Đề bài nhầm)

Giả sử đúng với $\left\lfloor \frac{n}{2} \right\rfloor$ tức là $T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \leq c \log \left\lfloor \frac{n}{2} \right\rfloor \leq c \log \frac{n}{2}$

Ta cần chứng minh $T(n) \leq c \log n$

Thay vào $T(n)$ ta có

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + 1 \leq c \log \frac{n}{2} + 1 = c \log n - (c - 1) \leq c \log n \text{ với mọi } c \geq 1$$

Vậy chọn $c = 2$ thì $T(n) \leq c \log n$ tức là $T(n) = O(\log n)$

b) $T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor + 12\right) + 1$ là $O(\log n)$ (NOTE : Đề bài nhầm)

Giả sử đúng với $\left\lfloor \frac{n}{2} \right\rfloor + 12$ tức là $T\left(\left\lfloor \frac{n}{2} \right\rfloor + 12\right) \leq c \log \left(\left\lfloor \frac{n}{2} \right\rfloor + 12\right) \leq c \log \left(\frac{n}{2} + 12\right)$

Ta cần chứng minh $T(n) \leq c \log n$ với một hằng số $c > 0$ nào đó

Thay vào $T(n)$ ta có

$$T(n) = T\left(\left\lfloor \frac{n}{2} \right\rfloor + 12\right) + 1 \leq c \log \left(\frac{n}{2} + 12\right) + 1 = c \log(n + 24) - (c - 1) \leq c \log n$$

$c \log \frac{n+24}{n} \leq c - 1$ khi n đủ lớn thì $\frac{n+24}{n} \rightarrow 1$ tức là về phải $c \log \frac{n+24}{n} \rightarrow 0$ do đó bất đẳng thức xảy ra khi $c \geq 1$ với n đủ lớn.

Ta chọn $c=2$ thì với $n \geq 24$ thì $T(n) \leq c \log n$ luôn đúng

Vậy $T(n) = O(\log n)$

c) Giải công thức đệ quy $T(n) = 2T(\sqrt{n}) + 1$

C1 : Ta dự đoán $T(n) = O(\log n)$

Giả sử đúng với \sqrt{n} tức là $T(\sqrt{n}) \leq c \log \sqrt{n} = \frac{c}{2} \log n$

Ta cần chứng minh $T(n) \leq c \log n$

$$Ta \text{ có } T(n) = 2T(\sqrt{n}) + 1 \leq 2 \cdot \frac{c}{2} \log n + 1 = c \log n + 1 \leq c \log n$$

Ta thấy điều này không xảy ra vì c là hằng số nên $c \log n + 1 > c \log n$

Liệu ta có cần tăng độ phức tạp của tiệm cận lên là $O(n)$?

Ta thấy giữa vế trái và vế phải của bất đẳng thức chỉ sai khác nhau một hằng số (là 1). Vì vậy thay vì giả sử ta giả sử $T(n) \leq c \log n$ ta giả sử $T(n) \leq c \log n - d$ với d là hằng số dương nào đó.

$$T(\sqrt{n}) \leq c \log \sqrt{n} - d = \frac{c}{2} \log n - d$$

$$T(n) = 2T(\sqrt{n}) + 1 \leq 2\left(\frac{c}{2} \log n - d\right) + 1 = c \log n - 2d + 1 = c \log n - d - (d - 1) \leq c \log n - d$$

Đúng với mọi hằng số $d \geq 1$. Ta chọn $c = 1$ và $d = 2$

$$T(n) \leq \log n - 2 \text{ Vậy } T(n) = O(\log n)$$

Kết luận: Khi sai khác chỉ 1 lượng nhỏ thì ta có thể dùng phương pháp trừ một hệ số có bậc nhỏ hơn vào hàm số dự đoán

C2: Ta dùng phương pháp đặt biến phụ

Đặt $m = \log n$ tức là $n = 2^m$

$$T(n) = T(2^m) = 2T(2^{m/2}) + 1$$

Đặt $S(m) = T(2^m)$

$$S(m) = 2S\left(\frac{m}{2}\right) + 1 = O(m)$$

Chú ý : để chứng minh $S(m) = O(m)$ thì ở đây ta cũng dự đoán $S(m) \leq cm - d$

Giả sử đúng với $\frac{m}{2}$ tức là $S\left(\frac{m}{2}\right) \leq c \frac{m}{2} - d$

Thay vào $S(m)$

$$S(m) = 2S\left(\frac{m}{2}\right) + 1 \leq 2\left(c \frac{m}{2} - d\right) + 1 = cm - d - (d - 1) \leq cm - d$$

Đúng với mọi giá trị của $d \geq 1$

Chọn $c = 1$ và $d = 2$ $S(m) \leq m - 2 = O(m)$

Vậy $T(n) = O(\log n)$

d) Tìm dạng hàm $g(n)$ đơn giản mà $f(n) = \Theta(g(n))$ cho các hàm $f(n)$ sau đây

- $f(n) = \sum_{i=1}^n \log i$

Ta có đẳng thức sau $\log n! = \Theta(n \log n)$

$$n! = \prod_{i=1}^n i$$

Lấy logarithm hai vế ta có

$$\log n! = \log \left(\prod_{i=1}^n i \right) = \sum_{i=1}^n \log i = \Theta(n \log n)$$

- $f(n) = \sum_{i=1}^n \frac{1}{i} = \Theta(\log n)$

Đây là một dạng của chuỗi số Harmonic (http://en.wikipedia.org/wiki/Harmonic_number)

$$f(n) = \sum_{i=1}^n \frac{1}{i} = H(n, -1)$$

$$\lim_{n \rightarrow \infty} H_n - \ln(n) = \gamma = 0.5772156649.. \text{ (Hằng số [Euler-Mascheroni](#))}$$

Một số công thức hay dùng

- $n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \left(1 + \Theta\left(\frac{1}{n}\right)\right)$
- $a^0 + a^1 + \dots + a^n = \frac{a^{n+1} - 1}{a - 1}$
- $\sum_{i=1}^n i c^i = c + 2c^2 + 3c^3 + \dots + n c^n = \frac{((n-1)c^{n+1} - n c^n + c)}{(c-1)^2}$ với $n \geq 0$, và $c \neq 1$
- $\sum_{i=1}^{\infty} x^i = \frac{1}{1-x}$ Nếu $|x| < 1$

Source code danh sách liên kết đơn

```
#include<stdlib.h>
#include<stdio.h>

typedef int DATA_TYPE; //dat DATA_TYPE la kieu int

typedef struct list {
    DATA_TYPE item;        /* data item */
    struct list *pNext;     /* point to successor */
} LIST;

void insert_list(LIST * &l, DATA_TYPE x)
{
    LIST *p;                /* temporary pointer */
    p = (LIST*)malloc( sizeof(LIST) );
```

```

        p->item = x;
        p->pNext = l;
    l = p;
}

//ham chen phan tu moi vao cuoi danh sach
void insert_last(LIST *l, DATA_TYPE x)
{
    LIST *p,*ptr;          /* temporary pointer */
    p = (LIST*)malloc( sizeof(LIST) );
    p->item = x;

    //Tim phan tu cuoi danh sach
    ptr=l;
    //neu danh sach rong
    if(ptr==NULL)
    {
        p->pNext=NULL;
        l=p;
    }
    else//tim phan tu cuoi cung (co con tro pNext la NULL)
    {
        while(ptr->pNext !=NULL) ptr=ptr->pNext;

        ptr->pNext=p;
        p->pNext=NULL;
    }
}

LIST *search_list(LIST *l, DATA_TYPE x)
{
    if (l == NULL) return(NULL);
    if (l->item == x)
        return(l);
    else
        return( search_list(l->pNext, x) );
}

//tra ve phan tu truoc phan tu can xoa
LIST *predecessor_list(LIST *l, DATA_TYPE x)
{
    if ((l == NULL) || (l->pNext == NULL)) {
        printf("Error: Danh sach rong hoac co 1 phan tu.\n");
        return(NULL);
    }
    if ((l->pNext)->item == x)
        return(l);
    else
        return( predecessor_list(l->pNext, x) );
}

void delete_list(LIST *l, DATA_TYPE x)
{
    LIST *p;          /* item pointer */
    LIST *pred;       /* predecessor pointer */
    p = search_list(l, x);
    if (p != NULL) {

```

```

        pred = predecessor_list(l, x);

if (pred == NULL)          /* splice out out list */
    l = p->pNext;
else
    pred->pNext = p->pNext;
    free(p);               /* free memory used by node */
    }
}

//Ham dung de in danh sach
void display(LIST *Head)
{
while(Head!=NULL)
    {
        printf("%i-->", Head->item);
        Head = Head->pNext;
    }
    printf("X\n");
}

//Ham dung de them cac phan tu vao cuoi day
void input_list(LIST *&Head)
{
    DATA_TYPE  x;
    printf("Nhap gia tri can them : ");
    scanf("%d", &x);
    insert_last(Head, x);
}

int main()
{
    LIST *Head=NULL;
    display(Head);

//them mot so phan tu vao dau danh sach
    insert_list(Head, 5);
    insert_list(Head, 3);
    insert_list(Head, 7);

    insert_last(Head, 12);
    insert_last(Head, 19);
    insert_last(Head, 14);

    input_list(Head);
    input_list(Head);

    display(Head);
    system("pause");
return 0;
}

```