HW 2

2-3

```
// a is array.  b is aim,  l, r is bound of search area.
void BinarySearchScope (int a[], int b, int l, int r, int &i, int &j){
    if (l>=r){
        i=r, j=l;
        return;
    }
    int mid = (l+r)/2 ;
    if (a[mid]==b){ i=j=mid;   return;}
    if(a[mid]<b)  l=mid+1;
    else  r=mid-1;
    BinarySearchScope(a, b, l, r, i, j);
}
```

2-4.

$$u \cdot v = (u_0 + u_1 \cdot 2^m + \cdots + u_k 2^{km}) v, \quad k = \lceil \frac{n}{m} \rceil.$$ $u_i (i=0,\cdots,k)$ 为以m位将 u "切割" 后的结果

$$= \sum_{i=0}^{k} u_k 2^{km} \cdot v$$

已知 $T(m \times m \text{ 位}) = O(m^{\log 3})$

$$\therefore T(uv) = kT(u_i \cdot v) = \lceil \frac{n}{m} \rceil \cdot O(m^{\log 3}) = O(nm^{\log 3 - 1}) = O(nm^{\log 3/2})$$

2-6.

$\because T(2^k \times 2^k) = O(7^k)$ (Strassen算法) $\quad T(m \times m) = m^3$

$\therefore T(n \times n) = T(m \times m) \times T(2^k \times 2^k) = O(7^k m^3)$

2-7

多项式为 $(x-n_1)(x-n_2)\cdots(x-n_d) = [(x-n_1)\cdots(x-n_{\frac{d}{2}})][(x-n_{\frac{d}{2}+1})\cdots(x-n_d)]$，设 $T(n)$ 为 n 个 $(x-n_i)$ 乘的时间

$$T(i)=\begin{cases} O(1), & i=1 \\ 2T(\frac{i}{2})+O(i\log i), & i>1 \end{cases}$$

不符合Master定理

由递归树: $T(d)=\frac{d}{2}\sum_{i=1}^{\log_2 d}\log\frac{1}{2^i} = \frac{d}{2}\log^2 d = O(d\log^2 d)$

2-13

将大于基准元素的置于其左边，小于等于基准元素的置于其右边