分治算法作业.

2.8

```
int l=1, r=n;
int find(int a[], int l, int r){
    int mid = (l+r)/2;
    如果 a[mid]==mid, 则 return mid.
    如果 a[mid]<mid. 则 find(a, mid+1, r).
    如果 a[mid]>mid, 则 find(a, l, mid).
}
```

2.9.

思维：求 T[0:n-1] 中的中位数x，并检验x的出现次数是否 > n/2.

```
Type Select(Type a[], int p, int r, int k){
    if (r-p<75){
        用 O(n^2) 简单排序对 a[p:r] 排序.
        return a[p+k-1].
    }
    for(int i=0; i<=(r-p-4)/5; i++){
        将 a[p+5i] - a[p+5i+4] 的第3小元素与 a[p+i] 交换位置.
    }
    Type x = Select(a, p, p+(r-p-4)/5, ((p+(r-p-4))/5+1)/2).
    int i = Partition(a, p, r, x), j = i-p+1;
```

```
if (k<=j). return Select (a,p,i,k);
else  return Select (a,i+1,r,k-j);
}.
```

Partition 函数为以x为基准，将小于x的元素放到x左边，大于x的元素放到x右边。

```
int Partition (Type a[], int p, int r){
    int i=p, j= r+1;
    Type x = a[p];
    while(1){
        while (a[++i] < x and i<r);
        while (a[--j] > x);
        if (i>=j) break;
        swap (a[i], a[j]);
    }
    a[p]=a[j];
    a[j]= x;
    return j.
}.
```

~~2.25 线性时间选择算法~~：

```
bool check(Type a[], int x){
    int cnt=0;
    for(int i=0; i<n; i++){
        if(a[i] == a[x]) cnt++;
    }
    if(cnt > n/2) return 1;
    else return 0;
}
```

主函数：

```
int m= Select(a, 0, n-1, n/2);
if(check(a, m)) T[0:n-1]有一个主元素.
else T[0:n-1]无主元素.
```

2-25.

解：假设输入元素被划分为7个一组.

共有$\lfloor \frac{n}{7} \rfloor$组.

则有. 至少有$4\lfloor \frac{n-7}{14} \rfloor$个比x大/小.

~~令 $4\lfloor \frac{n-7}{14} \rfloor \geq \frac{n}{4}$，解得~~ .

当 $n \geq 170$ 时，$4\lfloor \frac{n-7}{14} \rfloor \geq \frac{n}{4}$.

$$\therefore T(n) \leq \begin{cases} c_1, & n < 170 \\ c_2 n + T(\frac{n}{7}) + T(\frac{3n}{4}) & n \geq 170 \end{cases}$$

$\because \frac{1}{7} + \frac{3}{4} < 1$ $\therefore$ 时间复杂度仍为 $O(n)$，线性。

假设输入元素被划分为3个一组

共有 $\lfloor \frac{n}{3} \rfloor$ 组

则至少有 $2\lfloor \frac{n-3}{6} \rfloor$ ~~比x大~~ 个元素比 x 大/小。

当 $n \geq 40$ 时，$2\lfloor \frac{n-3}{6} \rfloor \geq \frac{n}{4}$

$$\therefore T(n) \leq \begin{cases} c_1, & n < 40 \\ c_2 n + T(\frac{n}{3}) + T(\frac{3n}{4}), & n \geq 40 \end{cases}$$

$\because \frac{1}{3} + \frac{3}{4} > 1$ $\therefore$ 时间复杂度不再是线性。