CSE 1201

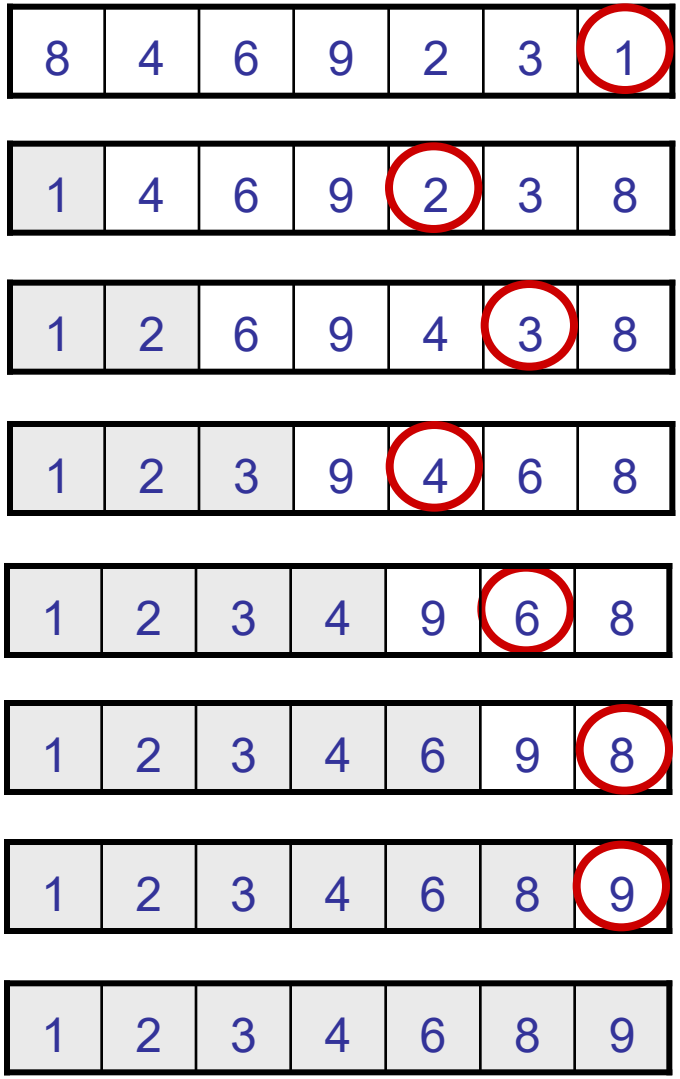
Selection Sort

Md. Shahid Uz Zaman
Dept of CSE, RUET

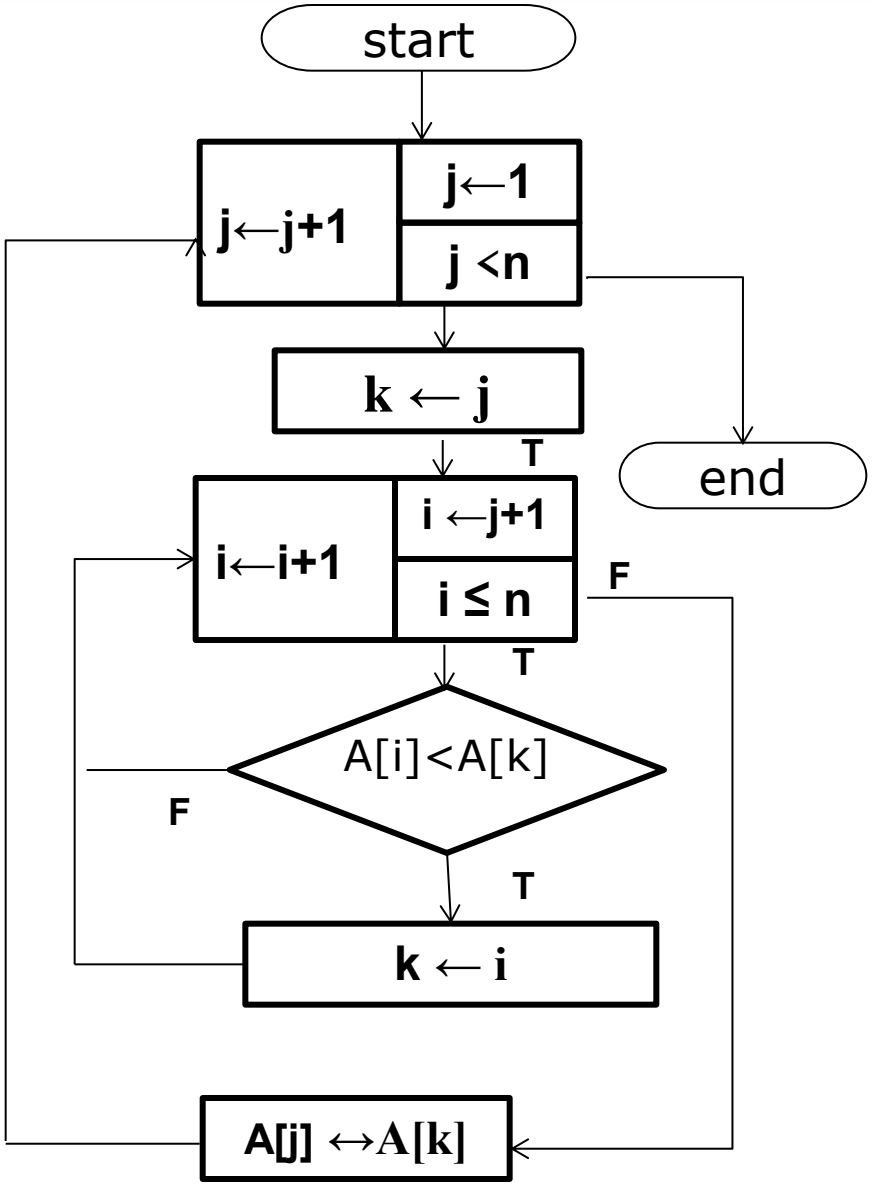
Selection Sort

- Idea:
  - Find the smallest element in the array
  - Exchange it with the element in the first position
  - Find the second smallest element and exchange it with the element in the second position
  - Continue until the array is sorted
- Disadvantage:
  - Running time depends only slightly on the amount of order in the file

Example

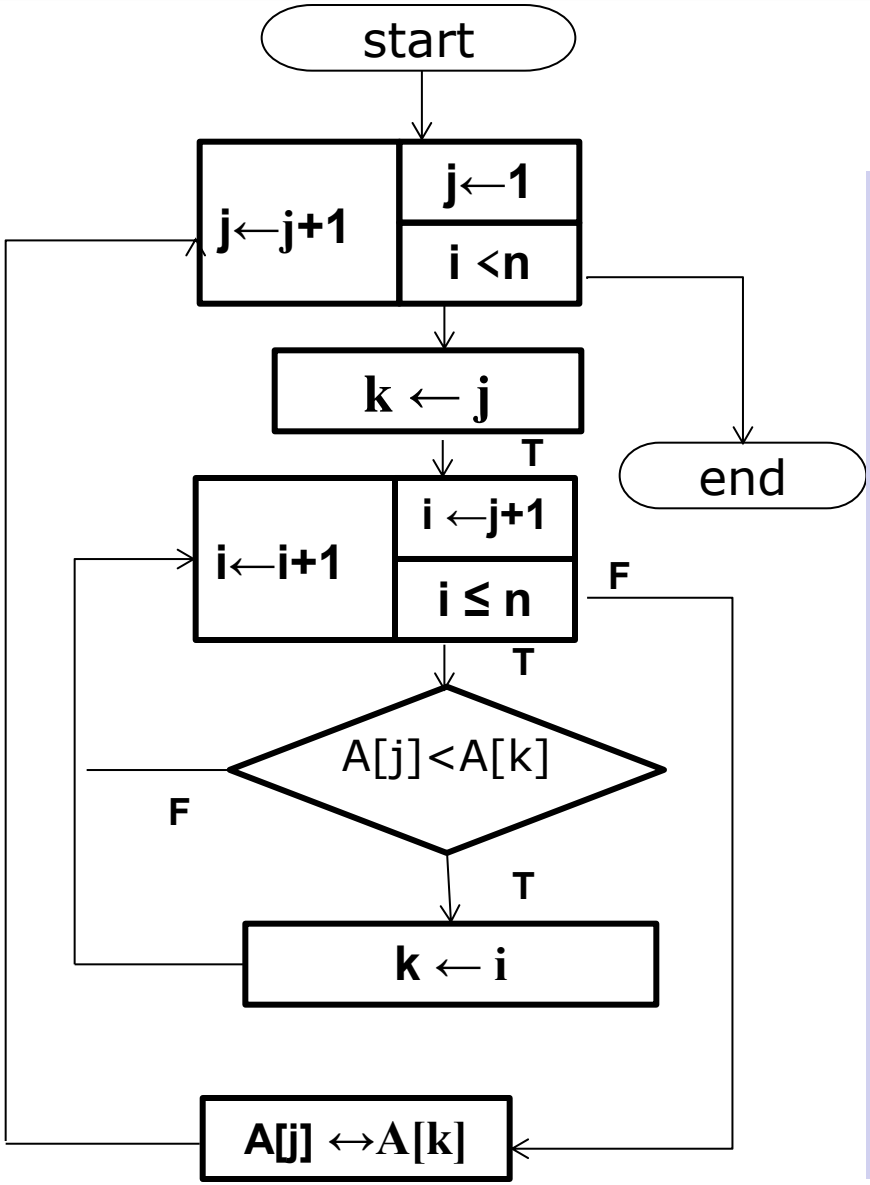

8 4 6 9 2 3 1
Pass 1: search (0 to n-1) lowest and swap with index 0
1 4 6 9 2 3 8
Pass 2: search (1 to n-1) lowest and swap with index 1
1 2 6 9 4 3 8
Pass 3: search (2 to n-1) lowest and swap with index 1
1 2 3 9 4 6 8
Pass 4: search (3 to n-1) lowest and swap with index 1
1 2 3 4 9 6 8
Pass 5: search (4 to n-1) lowest and swap with index 1
1 2 3 4 6 9 8
Pass 6: search (5 to n-1) lowest and swap with index 1
1 2 3 4 6 8 9
Pass 7: search (6 to n-1) lowest and swap with index 1
1 2 3 4 6 8 9

Selection Sort



8	4	6	9	2	3	1

Selection Sort




```
int main(){
    int i,j,k,t,n;
    int a[7] ={8,4,6,9,2,3,1};
    n=7;
    for(j=0;j<n-1;j++){
     k=j;
     for(i=j+1;i<n;i++)
         if(a[i]>a[k])
           k=i;
     t=a[j];a[j]=a[k];a[k]=t;
    }

    printf("The sorted list\n");
    for(i=0;i<n;i++)
     printf("%d ",a[i]);
}
```

Analysis of Selection Sort

Alg.: SELECTION-SORT(*A*)	cost	times
n ← length[*A*]	c_1	1
for j ← 1 **to** n - 1	c_2	n
do smallest ← j	c_3	n-1
for i ← j + 1 **to** n	c_4	$\sum_{j=1}^{n-1}(n-j+1)$
do if *A*[i] < *A*[smallest]	c_5	$\sum_{j=1}^{n-1}(n-j)$
then smallest ← i	c_6	$\sum_{j=1}^{n-1}(n-j)$
exchange *A*[j] ↔ *A*[smallest]	c_7	n-1

≈$n^2/2$ comparisons

≈n exchanges

$$T(n) = c_1 + c_2 n + c_3(n-1) + c_4\sum_{j=1}^{n-1}(n-j+1) + c_5\sum_{j=1}^{n-1}(n-j) + c_6\sum_{j=2}^{n-1}(n-j) + c_7(n-1) = \Theta(n^2)$$