Sorting (II): Selection Sort

CSCD 300 – Data Structures

Eastern Washington University

EASTERN WASHINGTON UNIVERSITY
start something big

Goal

We will learn the mechanism of the Selection Sort algorithm and then analyze its time complexity in the best as well as in the worst case.

Outline

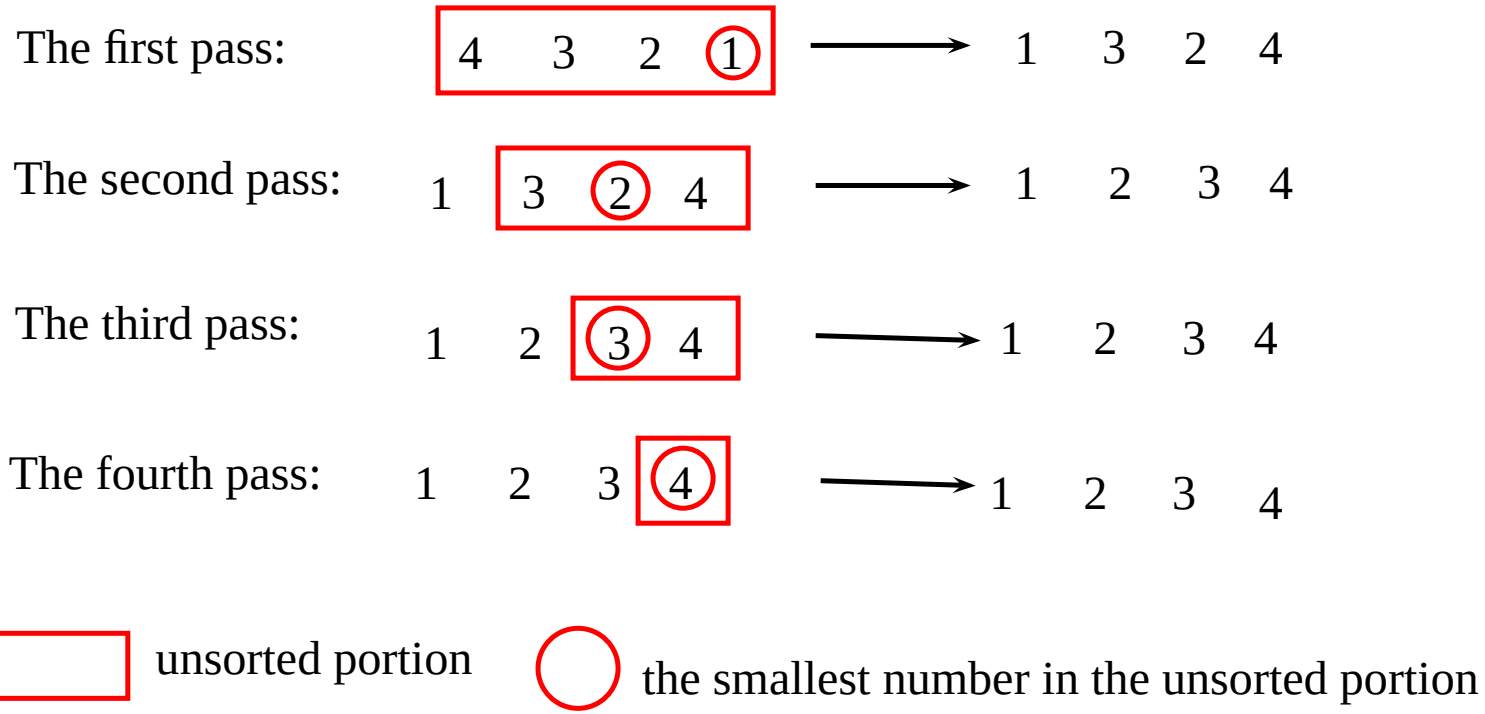
Selection sort

Basic idea

The algorithm repeatedly scans the unsorted portion of the sequence, and put the smallest number of the unsorted portion at the first location of the unsorted portion.

- The first pass is to scan $A[0 \ldots n-1]$, meaning the unsorted portion is the whole sequence.
- The second pass is to scan $A[1 \ldots n-1]$, meaning the unsorted portion is $A[1 \ldots n-1]$.
- ...
- The ith pass is to scan $A[i-1 \ldots n-1]$, meaning the unsorted portion is $A[i-1 \ldots n-1]$.
- ...

An example


The first pass:
4 3 2 1
1 3 2 4
The second pass:
1 3 2 4
1 2 3 4
The third pass:
1 2 3 4
1 2 3 4
The fourth pass:
1 2 3 4
1 2 3 4
unsorted portion
the smallest number in the unsorted portion

Pseudocode

SelectionSort(A)

Input: An array $A[0 \ldots n-1]$ of n numbers
Output: The sorted A.

```
for i = 0 ... n − 1 do
    min_index = i
    for j = i ... n − 1 do
        if A[j] < A[min_index] then
            min_index = j;
    exchange(A[i], A[min_index])
```

The time complexity

No matter what sequence of data is given:

- The first pass scans n elements.
- The second pass scans $n-1$ elements.
- ...
- The ith pass scans $n-i+1$ elements.
- ...

So the total time cost of the Selection sort in any case is:

$$n + (n-1) + (n-2) + \ldots + 2 + 1 = \frac{n(n+1)}{2} = O(n^2)$$

Question

How do you use Selection sort if the data sequence is saved in a singly linked list ?