Lesson 7: Sorting and Searching

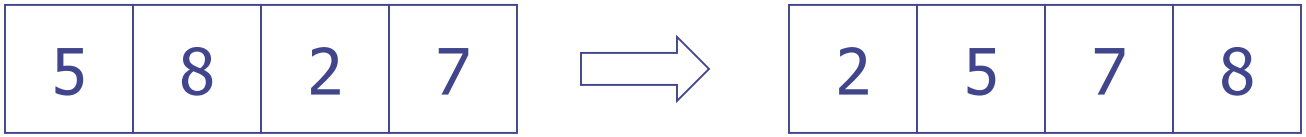
Jiun-Long Huang
Department of Computer Science
National Chiao Tung University

Algorithm

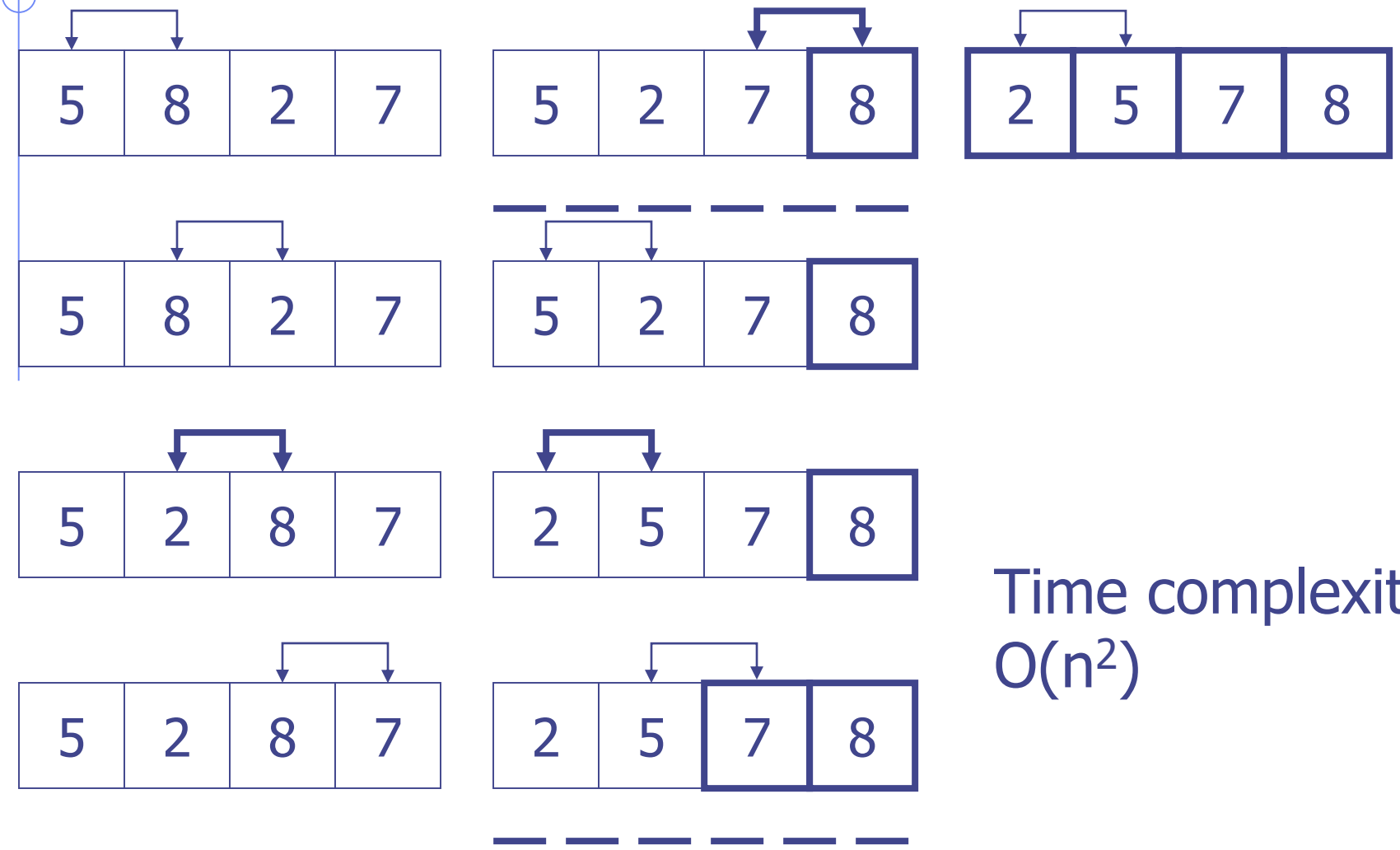
- Procedures to do some tasks
  - Ex. place some numbers in ascending order
- Measurements
  - Complexity: time complexity, space complexity

Sorting

- What is sorting?
  - Reordering records according to some attributes
  - According to scores
- Objective:
  - Speedup search



Bubble Sort

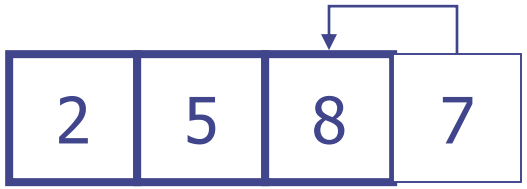


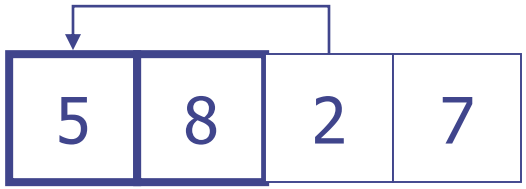
Time complexity
$O(n^2)$

```
#include <stdio.h>
int num[4];
void bubble(int itemNo)
{
  int i,j;
  int temp;
  for(i=itemNo-1;i>=0;i--)
    for(j=0;j<i;j++)
      if (num[j]>num[j+1])
      {
        temp=num[j+1];
        num[j+1]=num[j];
        num[j]=temp;
      }
}
```

```
int main(void)
{
  int i;
  printf("Input 4 numbers:");
  for(i=0;i<4;i++)
    scanf("%d",&num[i]);
  printf("Before sorting\n");
  for(i=0;i<4;i++)
    printf("%d ",num[i]);
  bubble(4);
  printf("\nAfter sorting\n");
  for(i=0;i<4;i++)
    printf("%d ",num[i]);
  return 0;
}
```

Insertion Sort





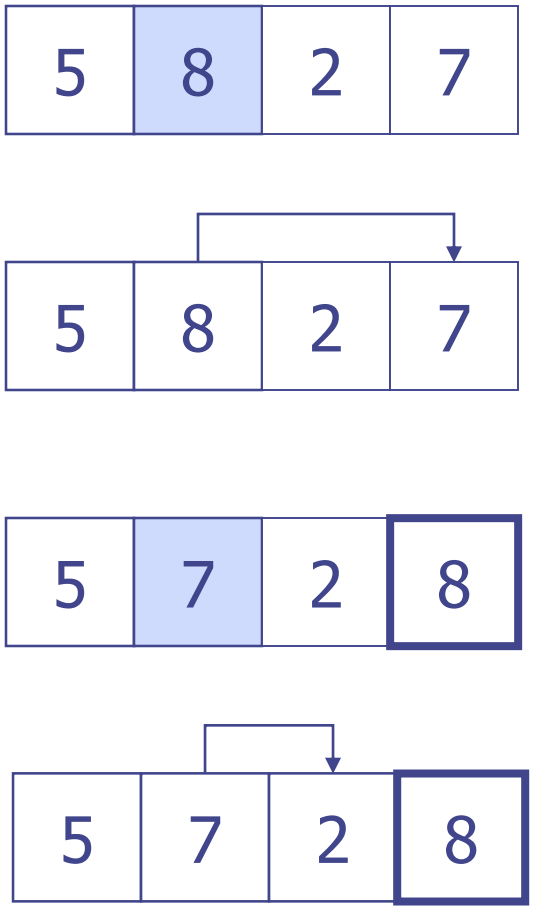
Time complexity
$O(n^2)$

```
#include <stdio.h>
int num[4];
void insertion(int itemNo)
{
  int i,j;
  int temp;
  for(i=1;i<itemNo;i++)
    for(j=i;j>0;j--)
      if (num[j]<num[j-1])
      {
        temp=num[j];
        num[j]=num[j-1];
        num[j-1]=temp;
      }
}
```

```
int main(void)
{
   int i;
   printf("Input 4 numbers:");
   for(i=0;i<4;i++)
     scanf("%d",&num[i]);
   printf("Before sorting\n");
   for(i=0;i<4;i++)
     printf("%d ",num[i]);
   insertion(4);
   printf("\nAfter sorting\n");
   for(i=0;i<4;i++)
     printf("%d ",num[i]);
   return 0;
}
```

Selection Sort

Time complexity
$O(n^2)$



```
#include <stdio.h>
int num[4];
void selection(int itemNo)
{
  int i,j,max_index;
  int temp;
  for(i=itemNo-1;i>=0;i--)
  {
    max_index=0;
    for(j=1;j<=i;j++)
      if (num[j]>num[max_index])
        max_index=j;
    temp=num[i];
    num[i]=num[max_index];
    num[max_index]=temp;
  }
}
```

```
int main(void)
{
   int i;
   printf("Input 4 numbers:");
   for(i=0;i<4;i++)
     scanf("%d",&num[i]);
   printf("Before sorting\n");
   for(i=0;i<4;i++)
     printf("%d ",num[i]);
   selection(4);
   printf("\nAfter sorting\n");
   for(i=0;i<4;i++)
     printf("%d ",num[i]);
   return 0;
}
```

Search

- Searching some records from massive data
  - Search for my examination sheet
- Exhaustive search (linear search)
  - Search all records
  - Time complexity: O(n)

Exhaustive Search(cont'd)

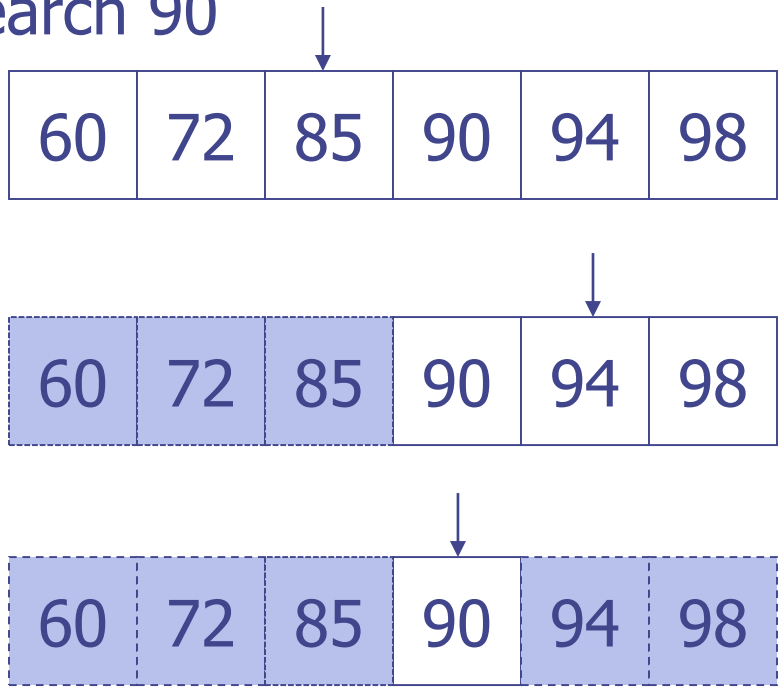
```
#include <stdio.h>
int num[10]={5,2,6,1,9,3,7,8,0,4};
int esearch(int value)
{
  int i;
  for(i=0;i<10;i++)
    if (num[i]==value)
      return i;
  return -1;
}
int main(void)
{
  int value;
  printf("Search for:");
  scanf("%d",&value);
  printf("In %d",esearch(value));
  return 0;
}
```

Binary Search

- Example:
  - When the examination sheets are sorted according to students' ids, how to efficiently find your sheet?
- Suitable for sorted data
  - Unsorted data→sort→binary search
- Time complexity: O(log n)

Binary Search

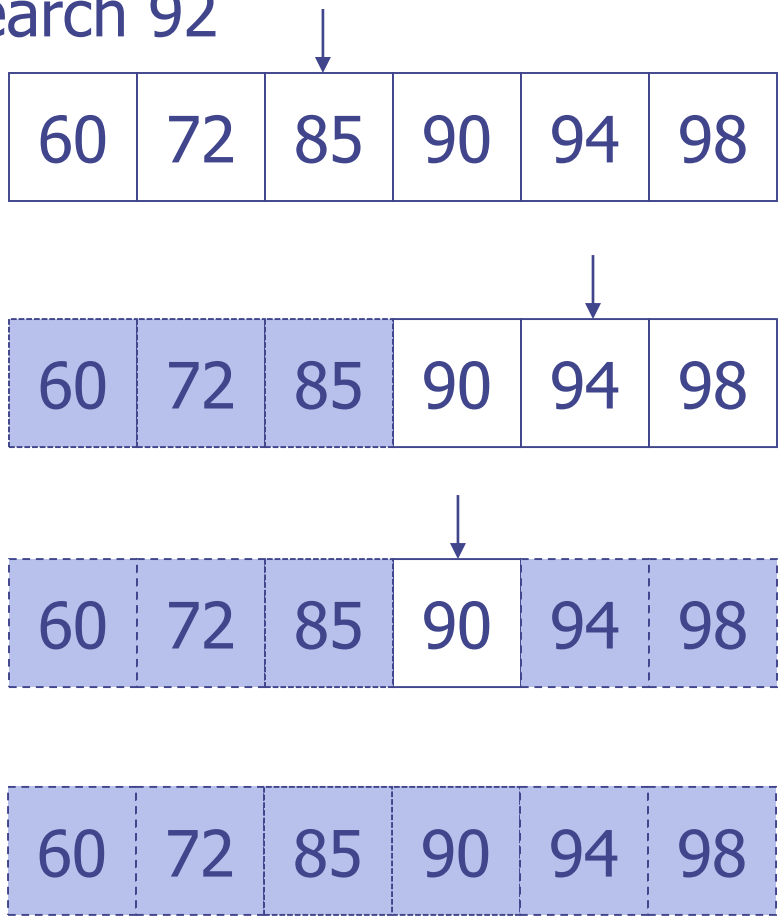
Search 90



left	right	middle
0	5	2
3	5	4
3	3	3

Binary Search (contd.)

Search 92



left	right	middle
0	5	2
3	5	4
3	3	3
4	3	

```
#include <stdio.h>
int num[10]={10,11,22,23,34,35,46,47,58,59};
int bsearch(int);
int main(void)
{
  int value;
  printf("Search for:");
  scanf("%d",&value);
  printf("In %d",bsearch(value));
  return 0;
}
```

```
int bsearch(int value)
{
  int left,right,middle;
  left=0;
  right=9;
  while(right>=left)
  {
    middle=(left+right)/2;
    if (num[middle]==value)
      return middle;
    else if (value>num[middle])
      left=middle+1;
    else
      right=middle-1;
  }
    return -1;
}
```