Data structure [B05]

김종규, PhD

2017-04-05

Review

- Linked list 로 구현 가능한 sorting algorithm
  - bubble sort ($O(n^2)$)
  - insertion sort ($O(n^2)$)
  - mergesort ($O(n \lg n)$, need extra memory)
  - Quicksort ($O(n^2)$, doubly linked list)
- Linked list 로는 구현 불가능한 sorting algorithm
  - Heapsort ($O(n^2)$, doubly linked list)

Outline

- Snapshot
- Snapshot sequence
- Illustrative examples
  - Partition (quicksort)
  - Heapify (heapsort)

Understanding an algorithm

```
def partition(A, p, r):
    x = A[r-1]
    i = p - 1
    for j in range(p,r-1):
        if A[j] <= x:
            i = i + 1
            A[i],A[j] = A[j],A[i]
    A[i+1],A[r-1] = A[r-1],A[i+1]
    return i+1
A = [0,1,5,2]
partition(A,0,len(A))
```

- How does the state change? ⟶ snapshot

Snapshot

- A state of a system
- The values of a variable when program execution paused
- By inspecting the change of snapshot, we could understand the algorithm better

Administrivia

- Quiz #02 4/12 (Wed)
- 중간고사 예행 연습
    - 지난 번 퀴즈보다 난이도 높음

Partition – 1

```
def partition(A, p, r):
  x = A[r-1]
  i = p - 1
  for j in range(p,r-1):
    if A[j] <= x:
      i = i + 1
      A[i],A[j] = A[j],A[i]
  A[i+1],A[r-1] = A[r-1],A[i+1]
  return i+1
A = [0,1,5,2]
partition(A,0,len(A))
```

```
A= [0, 1, 5, 2] p= 0 r= 4
x= 2 i= -1
i= -1 j= 0 A[j] <= x True
i= 0 j= 1 A[j] <= x True
i= 1 j= 2 A[j] <= x False
```

Partition – 2

```
def partition(A, p, r):
  x = A[r-1]
  i = p - 1
  for j in range(p,r-1):
    if A[j] <= x:
      i = i + 1
      A[i],A[j] = A[j],A[i]
  A[i+1],A[r-1] = A[r-1],A[i+1]
  return i+1
A = [0,1,2,5]
partition(A,0,len(A))
```

```
A= [0, 1, 2, 5] p= 0 r= 4
x= 5 i= -1
i= -1 j= 0 A[j] <= x True
i= 0 j= 1 A[j] <= x True
i= 1 j= 2 A[j] <= x True
```

Heapify

```
def heapify(A,i,heapsize):
    l = left(i)
    r = right(i)
    if l < heapsize
       and A[l] > A[i]:
        largest = l
    else:  largest = i
    if r < heapsize
       and A[r] > A[largest]:
        largest = r
    if largest != i:
        A[i],A[largest] = A[largest],A[i]
        heapify(A,largest,heapsize)
A = [0,1,3,5]
```

```
i= 3 A= [0, 1, 3, 5]
i= 2 A= [0, 1, 3, 5]
i= 1 A= [0, 5, 3, 1]
A= [5, 1, 3, 0]
```