Selection

Goldsmiths Computing

January 13, 2019

Motivation

- generalization of maximum operation
- component of solving real problems:
  - return the ten best matches to a query
  - return the median of this set of data

Definition

Selection is the operation of selecting the k^{th} largest element (with respect to some order relation) from a dataset of N elements.

Maximum

```
function MAXIMUM(A)
    result ← −∞
    for 0 ≤ i < LENGTH(A) do
        if A[i] > result then
            result ← A[i]
        end if
    end for
    return result
end function
```

Complexity analysis

- time: $\Theta(N)$
- space: $\Theta(1)$

Second

```
function SECOND(A)
    max ← −∞; result ← −∞
    for 0 ≤ i < LENGTH(A) do
        if A[i] > result then
            if A[i] > max then
                result ← max
                max ← A[i]
            else
                result ← A[i]
            end if
        end if
    end for
    return result
end function
```

Complexity analysis

- time: $\Theta(N)$ (but twice as much as for maximum)
- space: $\Theta(1)$ (but twice as much as for maximum)

kth

```
function KTH(A,k)
    maxes ← new collection(k)
    for 0 ≤ i < LENGTH(A) do
        if A[i] > SMALLEST(maxes) then
            REMOVE-MIN(A[i])
            INSERT(A[i],maxes)
        end if
    end for
    return MIN(maxes)
end function
```

Complexity analysis

maxes Array (unsorted)

- REMOVE-MIN is $\Theta(k)$
- INSERT is $\Theta(1)$
- REMOVE-MIN called $\Theta(N)$ times

$$\Longrightarrow \Theta(Nk)$$

kth

```
function KTH(A,k)
    maxes ← new collection(k)
    for 0 ≤ i < LENGTH(A) do
        if A[i] > SMALLEST(maxes) then
            REMOVE-MIN(A[i])
            INSERT(A[i],maxes)
        end if
    end for
    return MIN(maxes)
end function
```

Complexity analysis

maxes Array (sorted)

- REMOVE-MIN is $\Theta(1)$
- INSERT is $\Theta(k)$
- INSERT called $\Theta(N)$ times

$$\Longrightarrow \Theta(Nk)$$

kth

```
function KTH(A,k)
    maxes ← new collection(k)
    for 0 ≤ i < LENGTH(A) do
        if A[i] > SMALLEST(maxes) then
            REMOVE-MIN(A[i])
            INSERT(A[i],maxes)
        end if
    end for
    return MIN(maxes)
end function
```

Complexity analysis

maxes min-heap

- REMOVE-MIN is $\Theta(\log k)$
- INSERT is $\Theta(\log k)$
- each called $\Theta(N)$ times

$$\Longrightarrow \Theta(N \log k)$$

median

Selecting k$^{\text{th}}$ element takes $\Theta(N \log k)$ time

- selecting median ($\frac{N}{2}$th element) takes $\Theta(N \log N)$ time
- no better (asymptotically) than a full sort!

Can we do better?

- yes!
- quickselect, like partial quicksort
- compute the k$^{\text{th}}$ element in $\Theta(N)$ time (worst case)

Quickselect

```
function QUICKSELECT(S,low,high,k)
    if low = high then
        return S[low]
    else
        p ← PARTITION(S,low,high)
        if p = k then
            return S[k]
        else if k < p then
            return QUICKSELECT(S,low,p,k)
        else
            return QUICKSELECT(S,p+1,high,k)
        end if
    end if
end function
```

Median of medians

How to choose pivot for quickselect (and quicksort)?

- bad choice leads to $\Theta(N^2)$ (quadratic) performance

Guaranteed good choice of pivot for partitioning:

- break sequence into groups of 5
- compute the median of each group
- compute the median of the medians and use that as pivot

Recurrence relation

$$T(N) \leq T\left(\frac{N}{5}\right) + T\left(\frac{7N}{10}\right) + \Theta(N)$$

Can show by strong induction (or Akra-Bazzi method) that

$$T(N) \in \Theta(N)$$

Work

1. Reading:
   - CLRS, sections 9.1, 9.2
2. Questions from CLRS:
   - 9.1 Largest i numbers in sorted order