Question 7 - Kth Smallest Element Algorithm

Jaan Tollander de Balsch

2018-09-30

Given two sorted arrays $A[1...n]$ and $B[1...n]$, describe an algorithm that find the k-th smallest element in $A \cup B$ in $O(\log n)$ time.

The k-th element algorithm (1)

```
function kth(A, B, k)
    m = length(A)
    n = lenght(B)
    if m > n  # Let m <= n
        return kth(B, A, k)
    elseif m == 0
        return B[k]
    elseif k == 1
        return min(A[1], B[1])
    else
        h = div(k, 2)
        i = min(m, h)
        j = min(n, h)
        if A[i] > B[j]
            return kth(A, B[(j+1):end], k-j)
        else
            return kth(A[(i+1):end], B, k-i)
        end
    end
end
```

Input:

1) Sorted array $A = \langle a_1, ..., a_m \rangle$
2) Sorted array $B = \langle b_1, ..., b_n \rangle$
3) Integer $k \in \{1, 2, ..., m + n\}$

Output: k-th smallest element in $A \cup B$.

Recursive Case: The recursive step in the divide and conquer scheme for finding the k-th element works by excluding as many elements possible as from either array A or B that are quaranteed not to be the k-th smallest element. The algorithm achieves this by excluding

$$h = \lfloor k/2 \rfloor$$

elements per recursive step by choosing one element, a_i and b_j, from each array such that

$$i = \min(m, h)$$
$$j = \min(n, h)$$

By choosing the indices this way, will assure that the condition $i + j \leq k$ holds which is important for the k-th smallest element not to be excluded. The algorithm will then compare these two elements so that the correct half will be excluded.

a) If $b_j < a_i$ then exclude elements $\langle b_1, ..., b_j \rangle$ from the search. New inputs will be:
   1) $A_2 = A$
   2) $B_2 = \langle b_{j+1}, ..., b_n \rangle$
   3) $k_2 = k - j$

b) If $a_i \leq b_j$ then exclude elements $\langle a_1, ..., a_i \rangle$ from the search. New inputs will be:
   1) $A_2 = \langle a_{i+1}, ..., a_m \rangle$
   2) $B_2 = B$
   3) $k_2 = k - i$

Base Case:

a) If either array is empty the k-th smallest element is the k-th element in the not empty array because the array is sorted.

b) If $k = 1$ then the k-th smallest element is the minimum of elements a_1 and b_1

$$\min(a_1, b_1).$$

References

1. Kamath A. K-th element of two sorted arrays - geeksforgeeks. Available at: https://www.geeksforgeeks.org/k-th-element-two-sorted-arrays/ [Accessed October 1, 2018]