CS473-4

Boran Yildirim Duha Celik

April 2018

1

Assume that before $QUERY$, $PRE\_PROCESS$ is called and indexed array will be used for query. An example of code piece is as below.

$C[0..k] \leftarrow PRE\_PROCESS(A)$
$number \leftarrow QUERY(C, a, b)$

$PRE\_PROCESS(A)$
$C[0..k] \leftarrow$ new array
for i = 0 to k **do**
  $C[i] \leftarrow 0$
end for
for j = 1 to A.length **do**
  $C[A[j]] \leftarrow C[A[j]] + 1$
end for
for i = 1 to k **do**
  $C[i] \leftarrow C[i] + C[i-1]$
end for
return C

$QUERY(C, a, b)$
// Assume $a > 0$ and $b < k$.
if $a == 0$ **then**
  return C[b]
else
  return C[b] - C[a - 1]
end if

2

$SPARSE-TRANSPOSE(R, C, V, m, n, k)$
  $R'[0..n+1] \leftarrow$ new array
  $C'[0..k] \leftarrow$ new array
  $V'[0..k] \leftarrow$ new array

  fill R' with 0s

  for i = 0 to k **do**
    $R'[C[i]] \leftarrow R'[C[i]] + 1$
  end for

  cumsum $\leftarrow 0$
  for col = 0 to n **do**
    tmp $\leftarrow R'[col]$
    $R'[col] \leftarrow$ cumsum
    cumsum $\leftarrow$ cumsum + tmp
  end for

  for i = to m **do**
    for j = R[i] to R[i + 1] **do**
      col $\leftarrow C[j]$
      dest $\leftarrow R'[col]$
      $C'[dest] \leftarrow$ row
      $V'[dest] \leftarrow V[j]$
      $R'[col] \leftarrow R'[col] + 1$
    end for
  end for

  last $\leftarrow 0$
  for i = 0 to n + 1 **do**
    tmp $\leftarrow R'[i]$
    $R'[i] \leftarrow$ last
    last $\leftarrow$ tmp
  end for

  return R', C', V'

3

Pre :

```
struct element {
  int value;
  element * next;
}
int max = −∞;
for i = 0 to n do
  if A[i] > max then
    max ← A[i]
  end if
end for
element hash[max];
for i = 0 to n do
  element e ← new element;
  e.value ← i
  e.next ← hash[A[i]]
  hash[A[i]] ← e
end for
```

The Greatest element is found and indexing is done by it. Thus, collisions of different values in hashing are prevented. In a row there is only items whose values are same as their index(i). Insertion is done to head (last one in unsorted array is the head in the list).

Post :

```
for i = n - 1 to 0 do
  A[i] ← hash[A[i]].value
  hash[A[i]] ← hash[A[i]].next
end for
```

All elements from the sorted array replaced.