CMPS 6610 Problem Set 03

Answers

Name: Areen Khalaila

Place all written answers from `problemset-03.md` here for easier grading.

- **1b.**

Both the span and work are $O(n)$

- **1d.**

Work is $O(n)$ and span is $O(logn)$

- **1e.**

Work is $W(n) = W(n/3) + W(2n/3) + O(1)$ Therefore, it's $O(n)$

Span is $S(n) = S(2n/3) + O(1)$ Therefore, it's $O(\log n)$

- **2a.**

OR takes two booleans a and b and returns their logical OR

```
OR(a, b) =
  a   b
```

```
member(S, y) =
  reduce(OR,false,   S[i] = y: 0   i < |S|  )
```

```
dedup(A) =
  iterate(F,   , A)
  where F(out, y) =
      if member(out, y) then
        out
      else
        out ++  y
```

- **2b.**

```
combine(L, R) =
  L ++   R[j] : 0   j < |R| ,not member(L, R[j])
```

```
multi_dedup(A) =
  let
    B =  dedup(A[i]):0   i < |A|
  in
    reduce(combine,   , B)
  end
```

Work of member(out,y) is O(|out|).

Span of member(out,y) is O(log|out|)

In the worst case (all distinct), any order-preserving dedup built from iterate + member does

$\sum_{k=1}^{N} O(k) = O(n^2)$ work and

$\sum_{k=1}^{N} O(\log k) = O(n \log n)$ span

The worst-case asymptotics remain the same as Part 2a applied to a length N list:

Work: $O(n^2)$

Span: $O(n \log n)$

- **2c.**

Yes, we used iterate to process left-to-right and preserve first-appearance order. We also used reduce(OR) to implement member(out, y) in parallel.

- **3b.**

Work: $W(n) = W(n-1) + 1 \Rightarrow O(n)$

Span: $S(n) = S(n-1) + 1 \Rightarrow O(n)$

- **3d.**

$W_{map}(n) = O(n)$

$S_{map}(n) = O(n)$

$W_{scan}(n) = W_{scan}(n/2) + O(n) \Rightarrow O(n)$

$S_{scan}(n) = S_{scan}(n/2) + O(1) \Rightarrow O(\log n)$

$W_{reduce}(n) = 2W_{reduce}(n/2) + O(1) \Rightarrow O(n)$

$S_{reduce}(n) = S_{reduce}(n/2) + O(1) \Rightarrow O(\log n)$

Whole algorith:

$W(n) = O(n)$

$S(n) = O(\log n)$

- **3f.**

Work: $W(n) = 2W(n/2) + O(1) \Rightarrow O(n)$

Span: $S(n) = S(n) + O(1) \Rightarrow O(\log n)$