Assignment 3

Quinn Perfetto, 104026025
60-454 Design and Analysis of Algorithms

March 15, 2017

Question 1.

Idea: Sum consecutive elements of the input array until the sum exceeds M. Once this happens, add the offending index to the subdivision and reset the sum.

Algorithm 1: Subdivide(W, M)

Input: $W[1..n], 0 \leq W[i] \leq M, 1 \leq i \leq n$
Output: $S[1..k]$ such that S is an optimal subdivision of W

```
begin
    sum := 0;
    S := [ ];
    for i ← 1 to n do
        sum = sum + W[i];
        if sum > M then
            append(S, i - 1);
            sum := W[i];
        end
    end
end
```

Lemma 1.1. Algorithm Subdivide produces a valid subdivision of the input array W

We shall show this by inductively proving that after the mth iteration of the for loop,

$$S \text{ is a valid subdivision of } W[1..m] \wedge sum = \sum_{j=S_{last}+1}^{m} W[j]$$

Note: We take S_{last} to be the last element in S if it exists, and 0 otherwise.

Proof. (Induction Basis) We first note that sum is initialized to 0. After control reaches line 4 for the first time we have,

$$sum = sum + W[1] \Rightarrow sum = W[1] = \sum_{j=1}^{1} W[j]$$

Note that S was initialized to [ ]. Since $sum = W[1] \leq M$, control will not enter the if statement on line 5, thus S will remain empty and $S_{last} = 0$. Further since $W[1..m = 1]$ is a single element list such that $W[1] \leq M$, $S = [\ ]$ is vacuously a valid subdivision of W.

(Induction Hypothesis) Assume that after k iterations of the for loop,

$$S \text{ is a valid subdivision of } W[1..k] \wedge sum = \sum_{j=S_{last}+1}^{k} W[j]$$

(Induction Step) **Case 1:** $sum > M$

By the induction assumption S is a valid subdivision of $W[1..k]$, by the defintion of a valid subdivision we thus have,

$$\sum_{j=S_{last}+1}^{k} W[j] \leq M \qquad \text{(I)}$$

After appending $i - 1 = k$ to S, $S_{last} = k$. Therefore (I) is equivalent to,

$$\sum_{j=S_{last-1}+1}^{S_{last}} W[j] \leq M$$

Further since $\sum_{j=S_{last}+1}^{k+1} W[j] = W[k+1] \leq M$ we have S is a valid subdivision of $W[1..k+1]$.

After assigning $sum = W[k+1]$ we also have $sum = \sum_{S_{last}+1}^{k+1} W[j]$.

Case 2: $sum \leq M$

Since by our inductive assumption S is a valid subdivision of $W[1..k]$ and,

$$\begin{aligned} sum &= \sum_{j=S_{last}+1}^{k} W[j] + W[k+1] \\ &= \sum_{j=S_{last}+1}^{k+1} W[j] \\ &\leq M \end{aligned}$$

We have S is a valid subdivision of $W[1..k+1]$. $\square$

Therefore by Lemma 1.1, after n iterations S will be a valid parition of $W[1..n]$.
Hence the algorithm produces a valid subdivision of W.

Question 2 (b).

Algorithm 2: LeastDifferenceMatching(H, S)

```
Input: H = {h_j | 1 ≤ j ≤ n}, S = {S_j | 1 ≤ j ≤ m},  n ≤ m
Output: TODO
begin
    for i ← 1 to m do
        D[n + 1, i] = 0;
    end
    for i ← 1 to n − 1 do
        D[i, m] = ∞;
    end
    Sort(H);
    Sort(S);
    D[n, m] = |H[n] − S[m]| ;
    for i ← n to 1 do
        for j ← m - 1 to 1 do
            D[i, j] = min(|H[i] − S[j]| + D[i + 1,  j + 1],  D[i,  j + 1]);
        end
    end
end
```