CS 341 A3 Q5

Tareef Dedhar - 20621325

June 16, 2018

1 Shipping Paper

1.1 Description/Correctness

Sort the list of factories by the absolute value of vi - wi, or the marginal savings you would get by choosing the optimal warehouse for each factory. Then, greedily choose as much of the optimal warehouse's paper as possible for the first factory, and if you hit capacity for that warehouse fill the rest with the other warehouse. Repeat this until you've iterated through the entire list.

1.2 Pseudocode

```
int diff[n];
max_heap sorted[n][2]; // stores an index and the marginal difference in cost, sorted by the latter
int used_V[n]; // represents amount used from warehouse V for each factory Fi

for (int i = 0; i < n; ++i)
{
  int diff = |vi - wi|;
  sorted.insert(i, diff);
}

tuple factory = sorted.pop()
while (factory)
{
  if (vi > wi)
  {
    int temp = rV - di;

    if (temp >= 0)
    {
      rV -= di;
      used_V[factory.index] = di;
    }
    else
    {
      rW -= di - rV;
      used_V[factory.index] = rV;
      rV = 0;
    }
  }
  else wi > vi
  {
    int temp = rW - di;

    if (temp >= 0)
    {
      rW -= di;
      used_V[factory.index] = 0;
    }
    else
```

```
    {
      rV -= di - rW;
      used_V[factory.index] = di - rW;
      rW = 0;
    }
  }
}

return used_V;
```

1.3 Time Complexity

First, we initalize 2 empty arrays and 1 empty max heap. This should take constant time. Next, n times, we perform constant time subtraction and a log(n) time heap insert. This is a total of Θ(n logn) time.

Finally, we loop through the max heap of length n, each time performing a limited amount of integer comparisons and array/tuple acceses/writes, so constant time operations. This means we are again Θ(n logn), as each iteration is a log(n) heap pop, and constant time operations to populate the used_V output array.

1.4 Correctness

Assume for the sake of contradiction that for some greedy solution A, there exists a better optimal solution B. This means that there must be at least 1 factory in solution B where it uses more paper from the cheaper fatory. This means that for this factory, the greedy solution uses more of the more expensive factory than the optimal solution. However, by definition, the greedy solution could only have done so if there was another factory for which the difference in price per good was higher or equal to the cost of this factory using more of the worse choice. If the difference in the second factory was higher, than the greedy would be better than the optimal, which is impossible. So, this means they have to be equal, meaning that there cannot be a better optimal solution, as they must be equal.