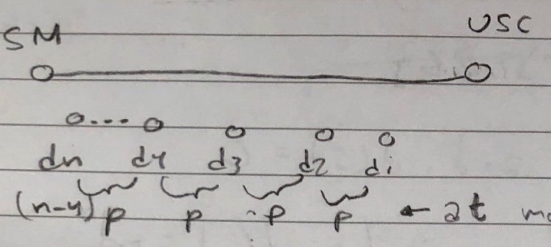
ESTO ES TIGRES

CSCI 570

Homework #4

1)



-Algorithm Proof, Time Complexity

Algorithm:

Assume the tank is full from beginning

Let D_{total} be the distance between USC and Santa Monica

Let $Tanks_{total}$ be the quotient of dividing D_{total} by p. $Tanks_{total}$ is the total amount of times the tank needs to get gas.

Let S be a max heap containing the gas stations with the longest distance from the previous gas station first. Thus, in decreasing order.

Let counter equal to zero.

Let Set_{final} be the set of gas stations that are needed to stop in order to get to Santa Monica

ESTO ES TIGRES

While counter not equal to $Tanks_{total}$

Get the first element of max heap S and store it in Set final

Add 1 to counter

End while

Return Set final

Time Complexity:

Calculate D_{total} - $O(1)$
Calculate $Tank_{total}$ - $O(1)$
Create S - $O(n)$
Create counter - $O(1)$
While loop - $O(n)$
Get the first element of S and maintain the heap for - $O(\log n)$
Store it in Set final - $O(1)$
Add 1 to counter - $O(1)$

$\therefore O(n + n + \log n) = O(2n + \log n) \approx O(n)$

Proof:

We use proof by contradiction. Suppose my algorithm is not optimal because we don't make as few gas stops as possible. Then let's say we choose another gas stop, instead of the first gas stop in the heap that has the greater distance between neighboring gas stations, which distance is $p/2$. We will call this gas station x. Let's assume

that the gas station between neighboring gas station is equal to p. We will call this gas station j. If the amount of gas station in Set final is k using my algorithm, then replacing j by x we will need at least to stop at another gas station in order to have enough gas to make it to Santa Monica. We will call this other gas station y, which will roughly have the distance p/2 between neighboring gas station, similar to x. If j is replaced by x and y, then instead of having k gas station stops, we will have k+1 gas stations. Since our goal is to make as few gas stops as possible along the way and $k < k+1$, then we would not be achieving our goal. This is a contradiction. Therefore, my algorithm must be true.

2) Does this modification work? Yes, it does work because Dijkstra's algorithm only positive values on the edges.

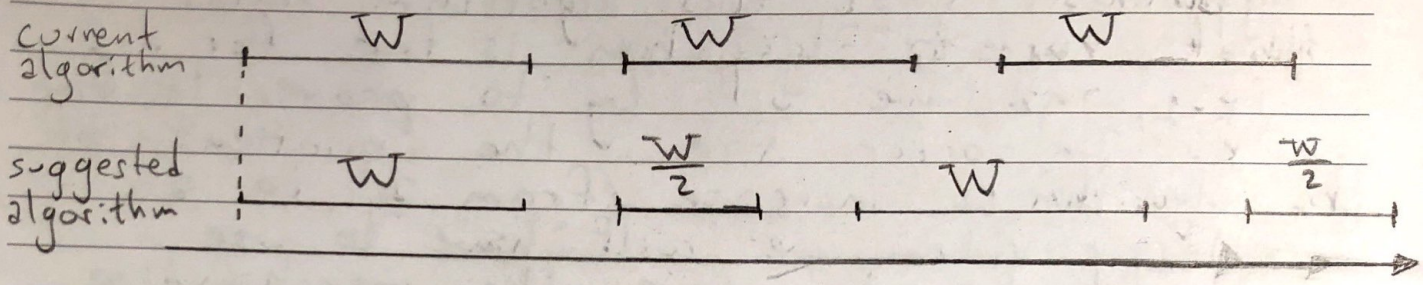
Proof: We use proof by contradiction. Suppose this modification does not work. Therefore, Dijkstra's algorithm could not be used in this situation to find the shortest path. However, Dijkstra's algorithm works with positive integers. In addition, we add w+1 to each edge length to make it positive. If Dijkstra's algorithm does not work, then at least an edge is negative. Thus, we have reached a contradiction from the problem statement. This is a contradiction,

ESTO ES TIGRES

Thus, Dijkstra's algorithm should work.

3) Chapter 4, Exercise 3

$$\sum w_i = W$$



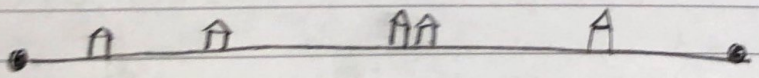
Our intuition for the greedy method came from wanting to minimize the amount of trucks and maximize the amount of weight of packages per truck. With the graph drawn above, it is easy to tell that the current algorithm is more efficient than the suggested algorithm, but we will prove that this holds for all other intervals possible. We will analyze the optimality of this greedy packaging algorithm by identifying the total trucks by total weights.

Proof: We will prove this statement by induction. Let A be my solution and O be an optimal solution. Let $i_1, \ldots, i_k$ be the set of trucks in A, where $|A| = K$. Let $j_1, \ldots, j_m$ be the set of trucks in O, where $|O| = m$. Our goal is to prove that $k = m$. In the graph, we have that the quotient of total trucks by total weight is equal to 1 in

the current algorithm. In contrast, the suggest algorithm yields to 4/3, which is bigger than 1. In order to find a better algorithm, this quotient would have to be less than 1. We will assume our induction hypothesis that this algorithm holds that current algorithm is true for i_{k-1}, and we will try to prove it for k. In order for the quotient of the algorithm to increase from 1 to a larger value, we will have to use more trucks to move the merchandise. This will cause the algorithm's k^{th} truck to "fall behind." But there's a simple reason why this could not happen: rather than choosing a truck with less weight on it, the greedy algorithm always has the option of choosing j_m to reduce the amount of trucks being used.

Furthermore, the quotient value cannot decrease from 1 (it cannot be less than 1) because trucks have a fixed limit W on the maximum amount of weight they are allowed to carry. Thus, fulfilling the induction step.

4) Chapter 4, Exercise 5



Solution: We will place a base station at the left of the first house we find and eliminate all the houses that are covered by

that station. Move four miles to the right and place another base station. Then, eliminate all the houses that are covered by the base stations. We will repeat this process until all houses are covered.