Tommy Vu

tommyuv23@csu.fullerton.edu

Project 2 - CPSC 335-01

Professor Bein

The Hypothesis

This experiment will test the following hypotheses:

1. Exhaustive search algorithms are feasible to implement, and produce correct outputs.
2. Algorithms with exponential running times are extremely slow, probably too slow to be of practical use.

Greedy Algorithm Code

greedy_max_weight(V, goods):	
todo = goods	O(1)
result = empty vector	O(1)
result_volume = 0	O(1)
while todo is not empty:	O(n)
Find the goods “a” in maximum weight per its volume	O(n)
Remove “a” from todo	O(c)
Let v be a’s volume in TEUs	O(1)
if (result_volume + v) <= V:	O(1)
result.add_back(a)	O(1)
result_volume += v	O(1)
return result	O(1)

Greedy Mathematical Analysis

Step 1: O(1 + 1 + 1 + n (n + c + 1 + 1 + 1 + 1) + 1)

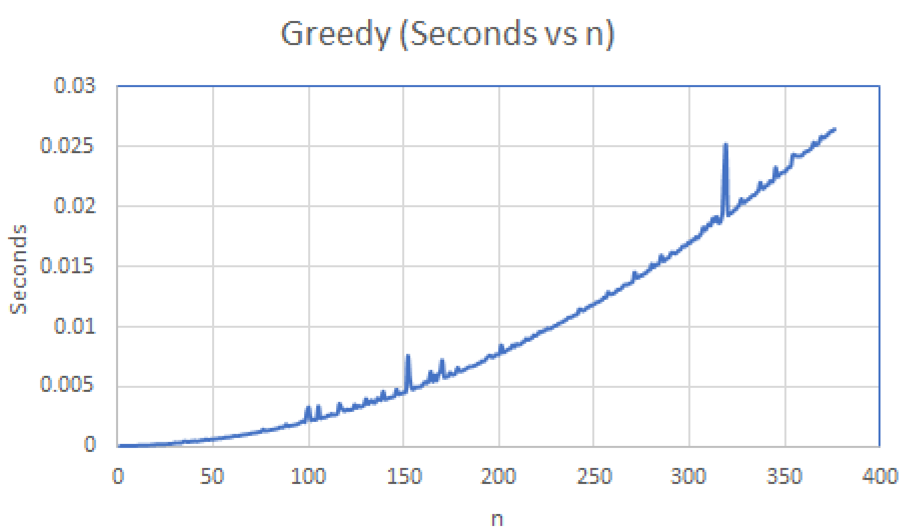
Step 2: O(4 + n(n + c + 4))

Step 3: O(4 + n^2+ nc + 4n)

Step 4: O(n^2 + 4n + nc + 4)

Answer = O(n^2)

Scatterplot



Exhaustive Algorithm Code

```
exhaustive_max_weight(V, goods):
        n = |goods||                                           O(1)
        best = None                                            O(1)
        for bits from 0 to (2^n -1):                           O(2 )
              candidate = empty vector                         O(1)
              for j from 0 to n-1:                             O(n)
                   if ((bits >> j) & 1) == 1:                  O(c)
                        candidate.add_back(goods[j])           O(1)


             if total_volume(candidate) <= V:                  O(1)
                 if best is None or                            O(1)
                  total_weight(candidate) > total_weight(best): O(1)
                 best = candidate                              O(1)
return best                                                    O(1)
```

Exhaustive Mathematical Analysis

Step 1: $O(1 + 1 + 2^n (1 + n(c + 1 + 1) + 1 + 1 + 1) + 1)$

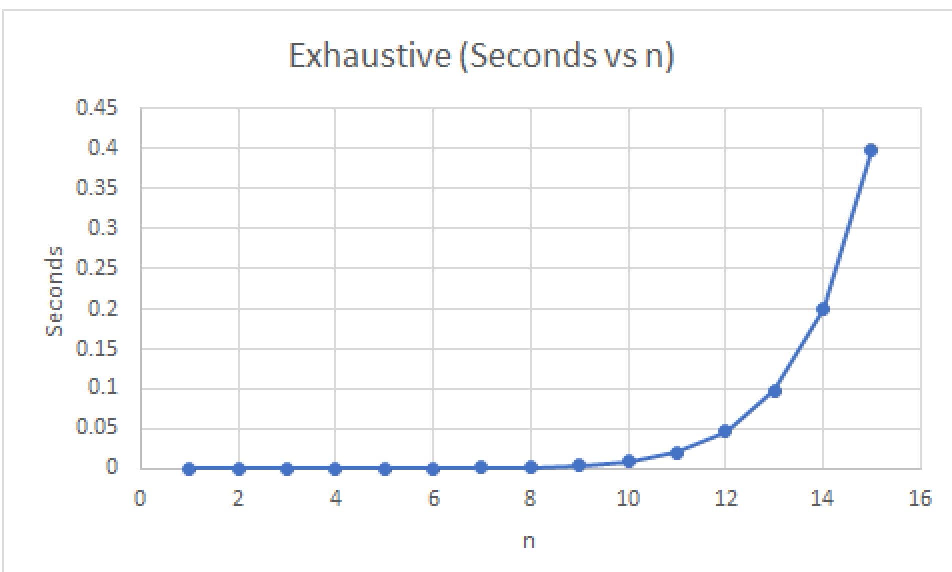
Step 2: $O(3 + 2^n (4 + n(c + 2))$

Step 3: $O(3 + 2^n (4 + nc + 2n))$

Step 4: $O(3^n + 8^n + 2^n nc + 4^n n)$

Answer = $O(2^n * n)$

Scatterplot



A) **Is there a noticeable difference in the performance of the two algorithms? Which is faster, and by how much? Does this surprise you?**

   There is a clear difference in the performance between the greedy algorithm and the exhaustive search algorithm. This is no surprise because the greedy algorithm looks for the most optimal path at each step, whereas the exhaustive search algorithm will look for the overall best result.

B) **Are your empirical analyses consistent with your mathematical analyses? Justify your answer.**

   Yes, the empirical analysis is consistent with the mathematical analysis because for greedy, we got a graph that showcases the $O(n^2)$ time complexity and for exhaustive, the graph matches that of an $O(2^n)$ graph.

C) **Is this evidence consistent or inconsistent with hypothesis 1? Justify your answer.**

   Yes, the evidence is consistent with hypothesis 1, as they are feasible to implement and they produce the correct outputs.

D) **Is this evidence consistent or inconsistent with hypothesis 2? Justify your answer.**

   The evidence is consistent with hypothesis 2 as well because as seen in the graph, as n gets larger, so does the time. This shows that algorithms with exponential running times are extremely slow and not practical to use.