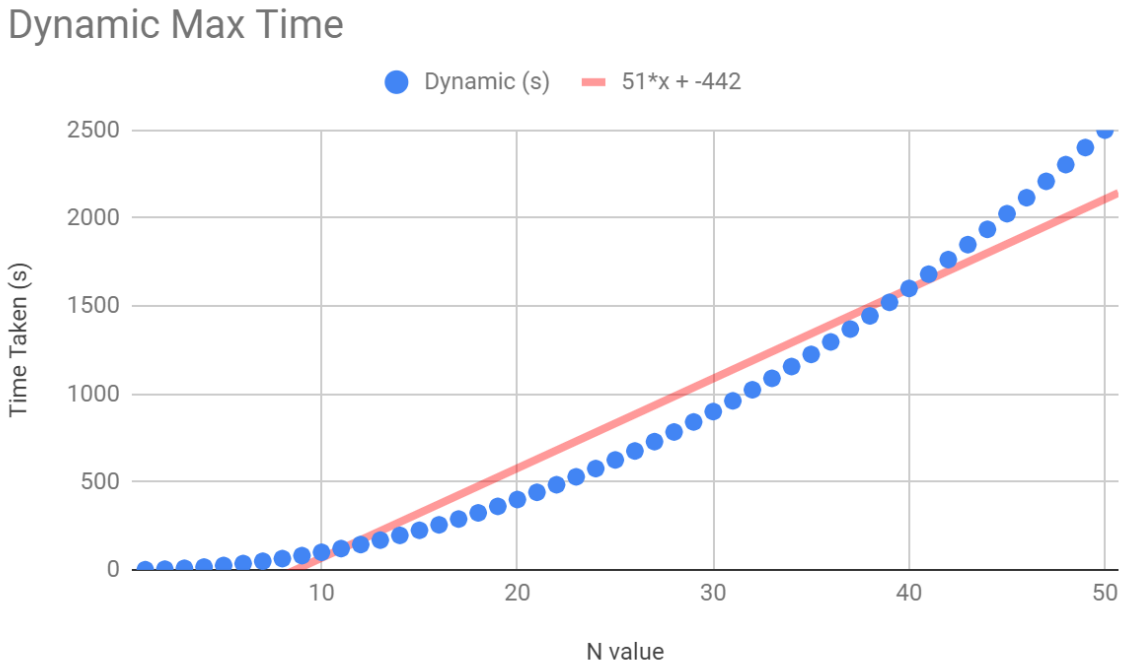
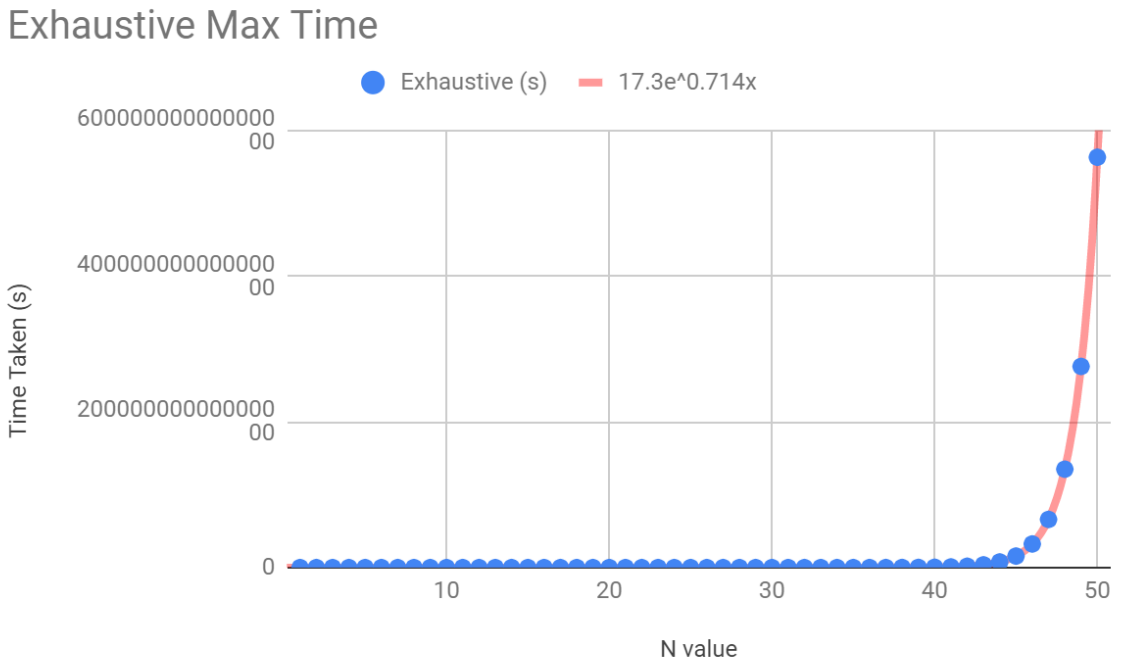
Project 4 Report

Yujin Chung yujin.chung@csu.fullerton.edu
Jason Liu jliu96@csu.fullerton.edu

SCATTERPLOT





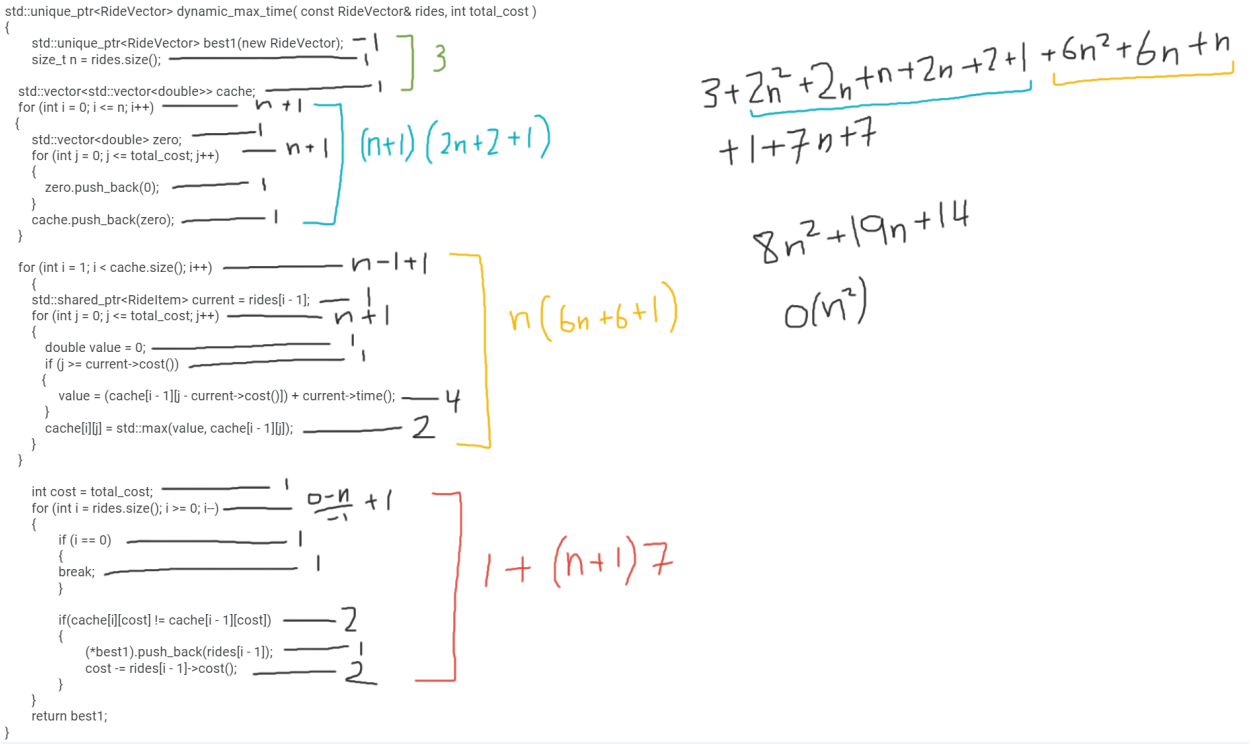
EMPIRICAL TIMING DATA GATHERED

N value	Dynamic (s)	Exhaustive (s)	Difference
1	1	2	1
2	4	8	4
3	9	24	15
4	16	64	48
5	25	160	135
6	36	384	348
7	49	896	847
8	64	2048	1984
9	81	4608	4527
10	100	10240	10140
11	121	22528	22407
12	144	49152	49008
13	169	106496	106327
14	196	229376	229180
15	225	491520	491295
16	256	1048576	1048320
17	289	2228224	2227935
18	324	4718592	4718268
19	361	9961472	9961111
20	400	20971520	20971120
21	441	44040192	44039751
22	484	92274688	92274204
23	529	192937984	192937455
24	576	402653184	402652608
25	625	838860800	838860175
26	676	1744830464	1744829788
27	729	3623878656	3623877927
28	784	7516192768	7516191984
29	841	15569256448	15569255607
30	900	32212254720	32212253820

31	961	66571993088	66571992127
32	1024	1.37E+11	1.37E+11
33	1089	2.83E+11	2.83E+11
34	1156	5.84E+11	5.84E+11
35	1225	1.20E+12	1.20E+12
36	1296	2.47E+12	2.47E+12
37	1369	5.09E+12	5.09E+12
38	1444	1.04E+13	1.04E+13
39	1521	2.14E+13	2.14E+13
40	1600	4.40E+13	4.40E+13
41	1681	9.02E+13	9.02E+13
42	1764	1.85E+14	1.85E+14
43	1849	3.78E+14	3.78E+14
44	1936	7.74E+14	7.74E+14
45	2025	1.58E+15	1.58E+15
46	2116	3.24E+15	3.24E+15
47	2209	6.61E+15	6.61E+15
48	2304	1.35E+16	1.35E+16
49	2401	2.76E+16	2.76E+16
50	2500	5.63E+16	5.63E+16

MATHEMATICAL ANALYSIS

Dynamic Programming Algorithm



Complexity = $O(n^2)$

Proof 1: $8n^2 + 19n + 14 \in O(n^2)$

Choose $c = |8| + |19| + |14| = 41$

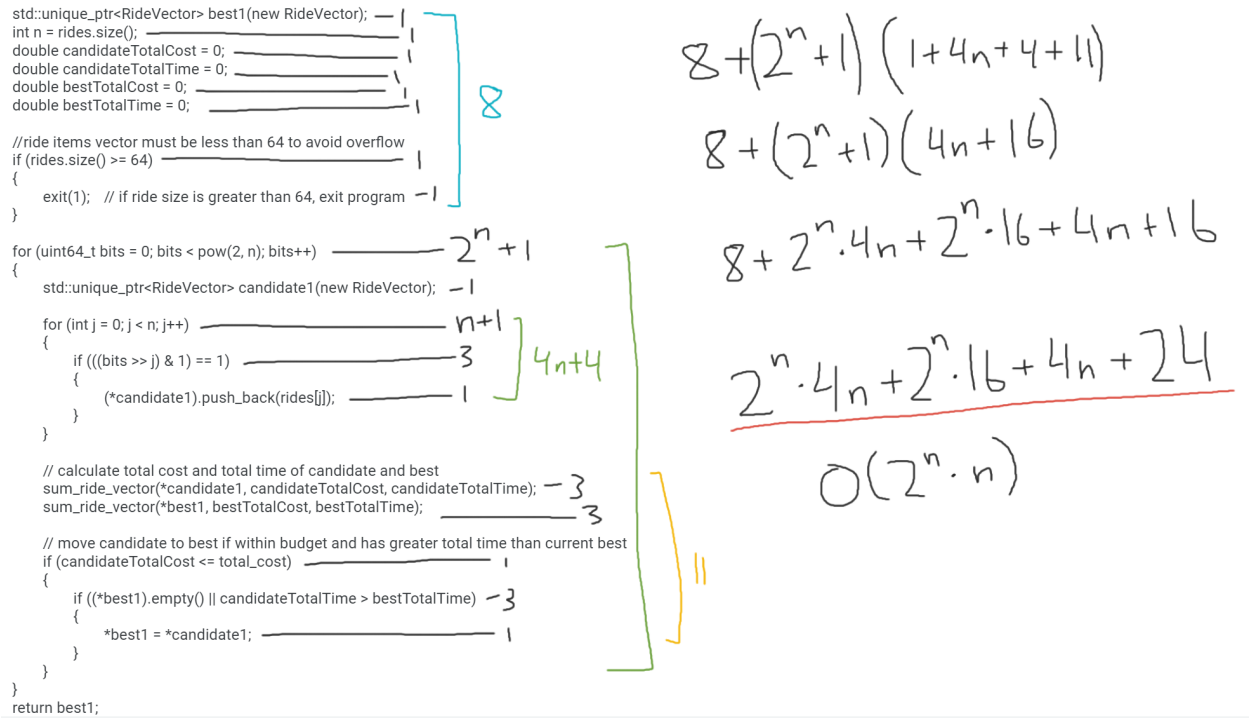
So $8n^2 + 19n + 14 \quad \forall \; n^2 \geq n_0$

$\Leftrightarrow 41n^2 - 8n^2 - 19n - 14 \geq 0$

$\Leftrightarrow 41 - 8 - 19 - 14 \geq 0 \quad \forall \; n \geq 1$

$0 \geq 0 \qquad\qquad n_0 = 1$

Exhaustive Search Algorithm



$8+(2^n+1)(1+4n+4+11)$

$8+(2^n+1)(4n+16)$

$8+2^n\cdot 4n+2^n\cdot 16+4n+16$

$2^n\cdot 4n+2^n\cdot 16+4n+24$

$O(2^n\cdot n)$

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

Proof 1: $2^n4n+ 2^n16 + 4n + 24 \in O(2^nn)$
Choose c = |8| + |32| + |4| + |24| = 68
So $2^n4n+ 2^n16 + 4n + 24 \quad \forall \ 2^nn \geq n_0$
$\Leftrightarrow 68*2^nn - 2^n4n - 2^n16 - 4n - 24 \geq 0$
$\Leftrightarrow 136 - 8 - 32 - 4 -24 \geq 0 \quad \forall \ n \geq 1$
$68 \geq 0 \qquad n_0 = 1$

QUESTIONS

Answers to the following questions, using complete sentences.

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

   Based on the graphs, there is a noticeable difference between the two algorithms. Dynamic 01 knapsack algorithm was a lot faster because its complexity is $O(n^2)$ while the exhaustive search algorithm is $O(2^n*n)$. It is difficult to say what the absolute difference between the two algorithms are because as n grows larger, exhaustive search will become slower.

b. Are your empirical analyses consistent with your mathematical analyses? Justify your answer.

   Our empirical analysis is consistent with the mathematical because the shape of our graph is the same as that of the mathematical analysis. For example, the dynamic algorithm calculation came out to $O(n^2)$ which is the same shape of our graph. Similarly, our empirical exhaustive search algorithm graph is exponential which is the same as the mathematical graph.

c. Is this evidence consistent or inconsistent with hypothesis 1? Justify your answer.

   With our program we were able to prove that it is possible to implement an exhaustive search algorithm that also produces outputs that are correct. The outputs for both dynamic and exhaustive were the exact same.

d. Is this evidence consistent or inconsistent with hypothesis 2? Justify your answer.

   After calculating the run time of exhaustive search it is clear that at a value of roughly n=5 the amount of time it takes would be extremely impractical to use in real world applications.