Offline 1

Name : Md Sabbir Rahman

Roll : 1705076

Section : B1

Course : CSE 204, Data Structure and Algorithms Sessional

Submission Date : 10-06-2019

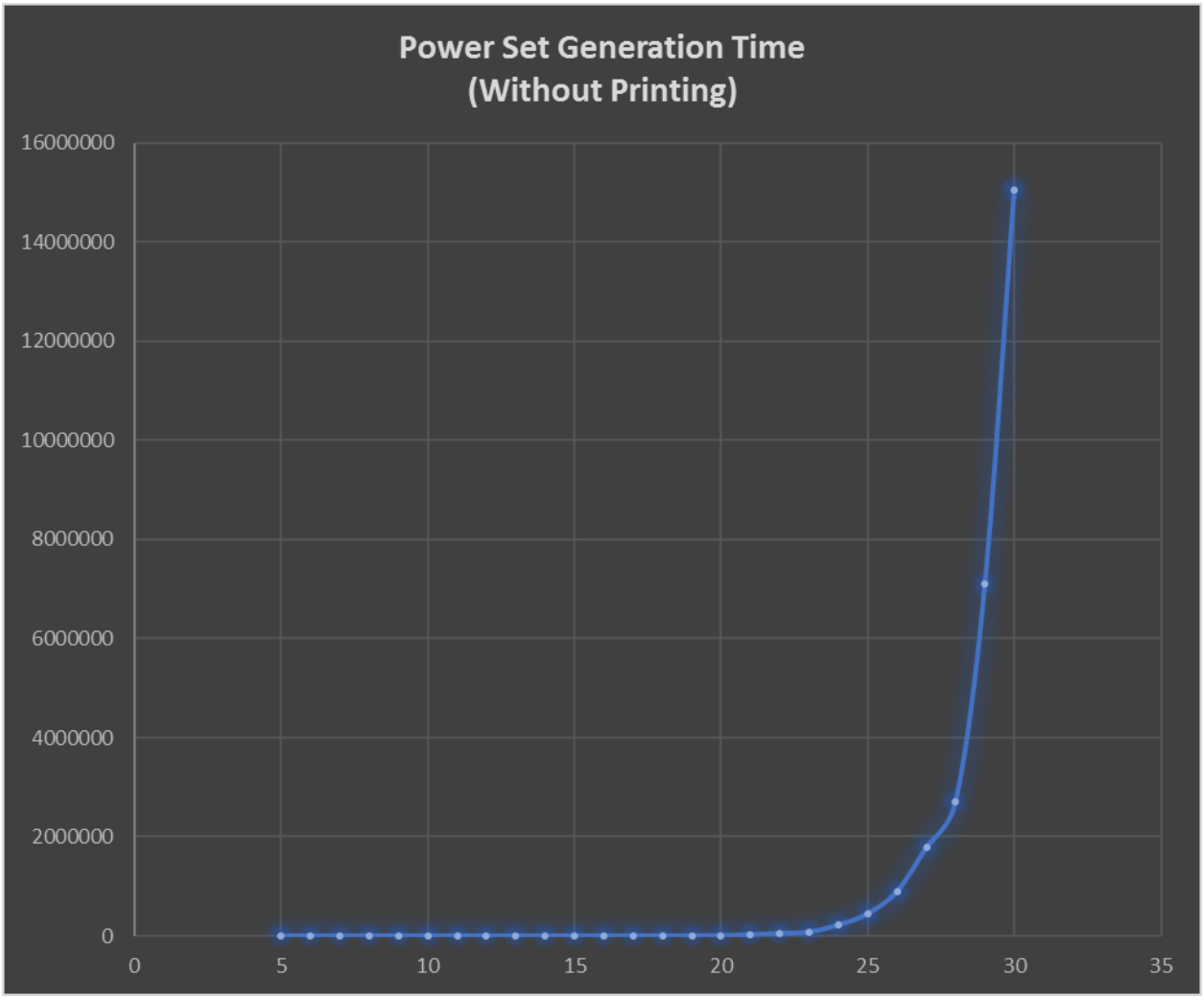
Objective: The objective is to run insertion sort and selection sort on arrays of different sizes and compare their worst case, best case and average case runtime.

Machine Specs: Intel Core i5-4460 CPU @ 3.20 GHz, 8.00 GB, x64 based processor. Windows 8.1 Operating System.

Data: In the enclosed cpp file, Power set generation algorithm was implemented using recursive backtracking and run on randomly generated array of various sizes (after sorting). The set is generated using random characters from 0-9, a-z, A-Z. For the Algorithm , the sets were first sorted and then the backtracking was executed. The execution time has been added as a excel file with the report. We notice that the algorithm is very slow and can only work upto set of size 30 under 10 seconds. The algorithm shows exponential complexity.

Set Size	Power Set Generation Time (Without Printing)
5	0.96
6	0.96
7	2.56
8	3.21
9	7.05
10	14.11
11	27.26
12	55.81
13	110.01
14	217.78
15	434.93
16	868.56
17	1321.7
18	3572.7
19	5427.9
20	10488
21	27962
22	56437
23	81838
24	225036
25	450849

26	901204
27	1783541
28	2709476
29	7105348
30	15052338



Complexity Analysis: In the worst case all elements of the set are distinct, so there are n distinct elements. The recursion will have running time T(k) = T(k-1) + T(k-1). In the base case, T(0) = O(n) as all elements need to be checked whether to include it in subset or not. The resulting complexity is T(n) = O($n \cdot 2^n$).

Discussion: The running time were measured by using windows' QueryPerformanceCounter function which is very accurate. The time was measured before calling and after power set

generation to only measure execution time of the function. With even bigger inputs, The actual complexity of the functions would become more clear.