ICDPIC Patient Lookup D-Ary Heap Optimization

Shariq Butt*

*shariqbutt@icloud.com

ABSTRACT

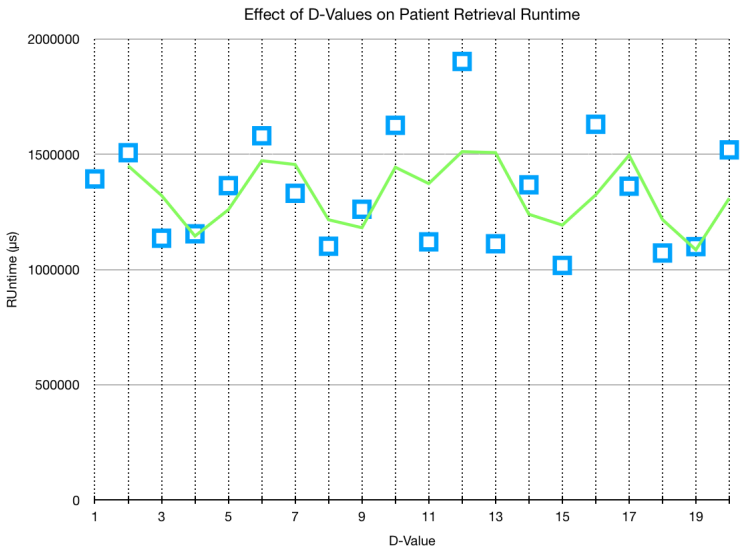
The ICD is the foundation for the identification of health trends and statistics globally. It is the international standard for defining and reporting diseases and health conditions. It allows the world to compare and share health information using a common language. The ICDPIC uses R statistical software to translate common codes from the ICD-9 and ICD-10 to more usable and relevant clinical data. The specific lookup table used in this experiment was the ICD-10-ROC, which maps ICD-10-CM codes to injury severity and body region using a prediction model trained on the National Trauma Bank. Through implementing a max d-ary heap, we were able to make a patient lookup and retrieval for the given data. Data was collected through compiling in order to initialize data and then testing retrieval of the most injured patient. Run-times were collected in microseconds and d-values were tested 3 times and averaged in order to obtain more accurate data and negate the effects of background programs such as music players, RuneScape instances, and Game Boy Advance emulators. The data collected indicates that the optimum d-value for the max d-ary heap is 15.

Introduction

The purpose of this experiment was to implement patient lookup with a priority queue using a d-ary max heap. The algorithm is meant to take patient data and then index them into a heap, and through the pop() function of the heap allows for the lookup and retrieval of the patient with the most severe injuries who requires the most attention. Through using a priority queue, the heap should be able to perform functions of popping and inserting at $O(d * log_d(n))$ and $O(log_d(n))$ times, respectively, and the assignment of priority is dictated by the injury severity score of the patient. The data used is not actual hospital data but instead simulated data from a predictive model, and should serve to accurately represent the use case.

Data

d	runtime (µs)
1	1392885
2	1506625
3	1135626
4	1154794
5	1363852
6	1580144
7	1330612
8	1100966
9	1261455
10	1625582
11	1119967
12	1902521
13	1111641
14	1367355
15	1017777
16	1630047
17	1361347
18	1071574
19	1099303
20	1518968



Conclusion

The data indicates that 15 is the optimum d-value for the data structure, with 18 coming close enough to be within the margin of error. However, even with this d-value optimization, the run-time still leaves something to be desired. Upon examining the data, this lack of speed is understandable, as patient data has many cases which have low severity and duplicate cases. This algorithm would be better suited to somewhere like an ICU, which has more severe cases and less duplicates. In a setting like a general hospital or clinic, there are too many duplicates and as these duplicate cases grow in number the purpose of the d-ary heap data structure is defeated, and the use case becomes better suited for something like a binary search tree.