CSE: 224 DATASTRUCTURE & ALGORITHMS

Lab 5 Report

Names	IDs
Aaser Fawzy Zakaria Hassan	19015403
Mohamed Ezzat Saad El-Shazly	19016441
Ahmed Hamdy Ahmed Osman	19017253

Contents

Problem Statement:

1. You are required to implement a generic B-Tree where each node stores key-value pairs and maintains the properties of the B-Trees use the provided interfaces.

2. You will be given a set of Wikipedia documents in the XML format, and you are required to parse them and maintain an index of these documents content using the B-Tree to be able to search them efficiently use the provided interfaces.

Requirements:

1. A generic B-Tree data structure that implements the given interfaces.
2. A Search Engine that uses the B-tree to search for given words in the provided Wikipedia xml documents.

Implementation details & design choices:

B- Tree

1- using 3 lists to store keys, values and children's in every node.

2- to enter any key or value to node it should be wrapped in a list

3- key enter first then value or child second in the node.

4- the operations (insertion & deletion) happens first and if there is error fixup take place second.

5- search return null if the object is not in the tree.

6- in order to make the mapping there is a control variable in the code the node where the key then value then value or the children second.

7- we use inline coding to make the code faster

8- transvers done using stack and iterative method no recursive

---

Search Engine

- Based on the above implementation the Search engine first indexes the files/folders then the read values are then inserted with key of type Integer: id of the "doc" and with value: context of the "doc" into the webpage tree.

- The searching for one word traverses each node in the B-Tree using recursive in-order traversal to search each node's list of values for the word then counting its frequency in this string. The result is then added to ISearchResult and pushed into the list.

- The searching of multiple words assuming they are separated by " " are split into multiple words and searched for individually and then for each result there will be a list, and for each "doc" searched in the word with the minimum frequency is picked. After this the results are added as above.

- The string for searching for multiple words should be " " separated.

Analysis: Graphs & Tables

B-Tree:

Tables:

Insertion table

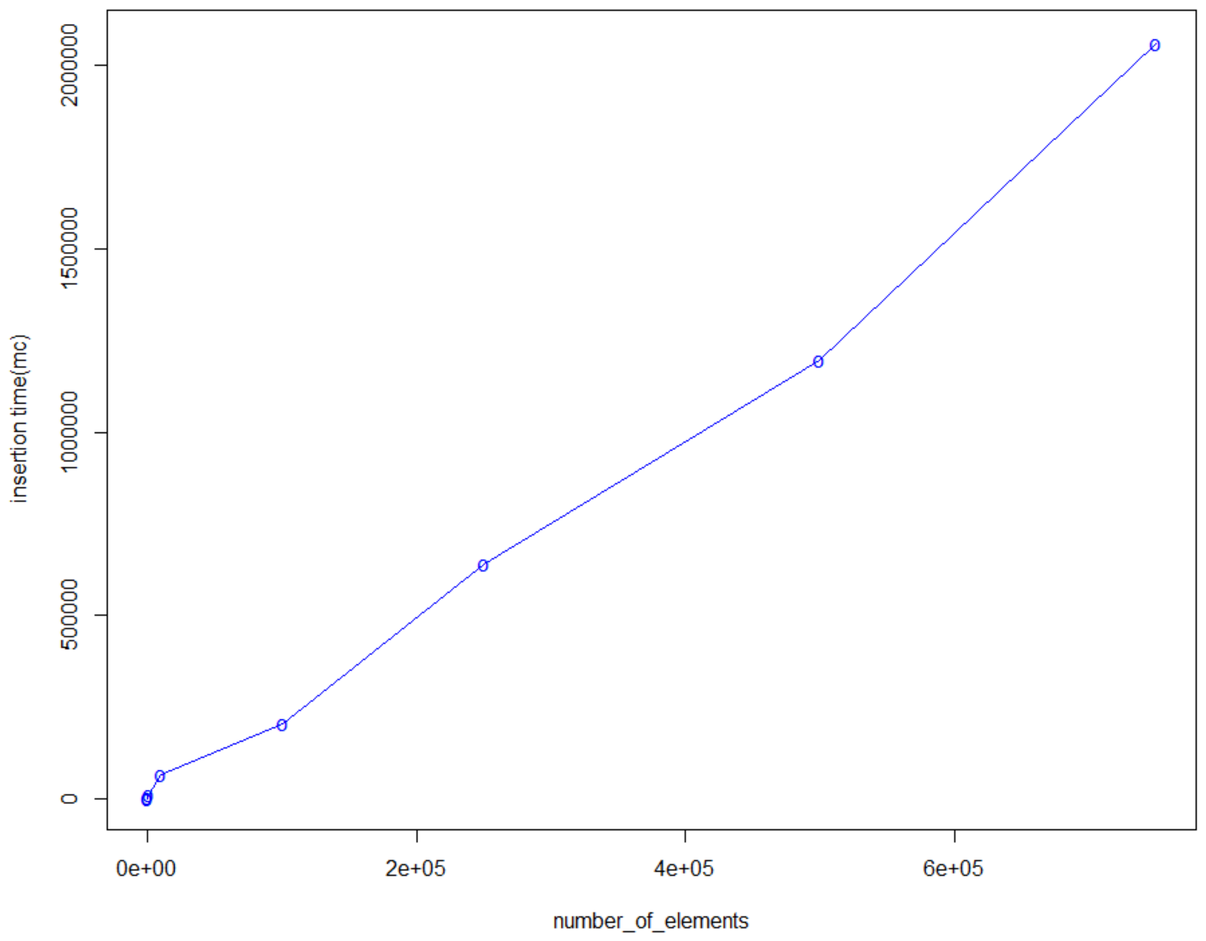
N/trial	0	1	2	3	4	5	6	7	8	9	Average
10	875900.0	230200.0	145500.0	156500.0	227500.0	159400.0	151700.0	197200.0	134300.0	130600.0	240880.0
100	5555500.0	2645300.0	2863100.0	3951800.0	1366200.0	2368200.0	983700.0	886000.0	646300.0	779900.0	2204600.0
1,000	26360300.0	5912600.0	17073700.0	11326000.0	13359100.0	5462200.0	5632800.0	13566400.0	15161300.0	10144800.0	12399920.0
10,000	178165100.0	74769400.0	44985100.0	21459100.0	20859200.0	36777500.0	154466700.0	52386800.0	32202700.0	24782500.0	64085410.0
100,000	610810000.0	235832500.0	185434200.0	148511300.0	135870700.0	169909500.0	121102400.0	130757800.0	151063100.0	146077500.0	203536900.0
250,000	1736296300.0	533470800.0	601115500.0	606428400.0	530429800.0	655777700.0	431614900.0	404644200.0	514957000.0	402703900.0	641743850.0
500,000	2414155900.0	1297792900.0	1358346100.0	1068822400.0	1057160400.0	986078200.0	860280900.0	960560100.0	1008463900.0	943955100.0	1195561590.0
750,000	3099683800.0	2173296300.0	1888524900.0	1564157600.0	1858084400.0	1659904900.0	2189846000.0	2130988100.0	2073710200.0	1968404400.0	2060660060.0

Deletion table

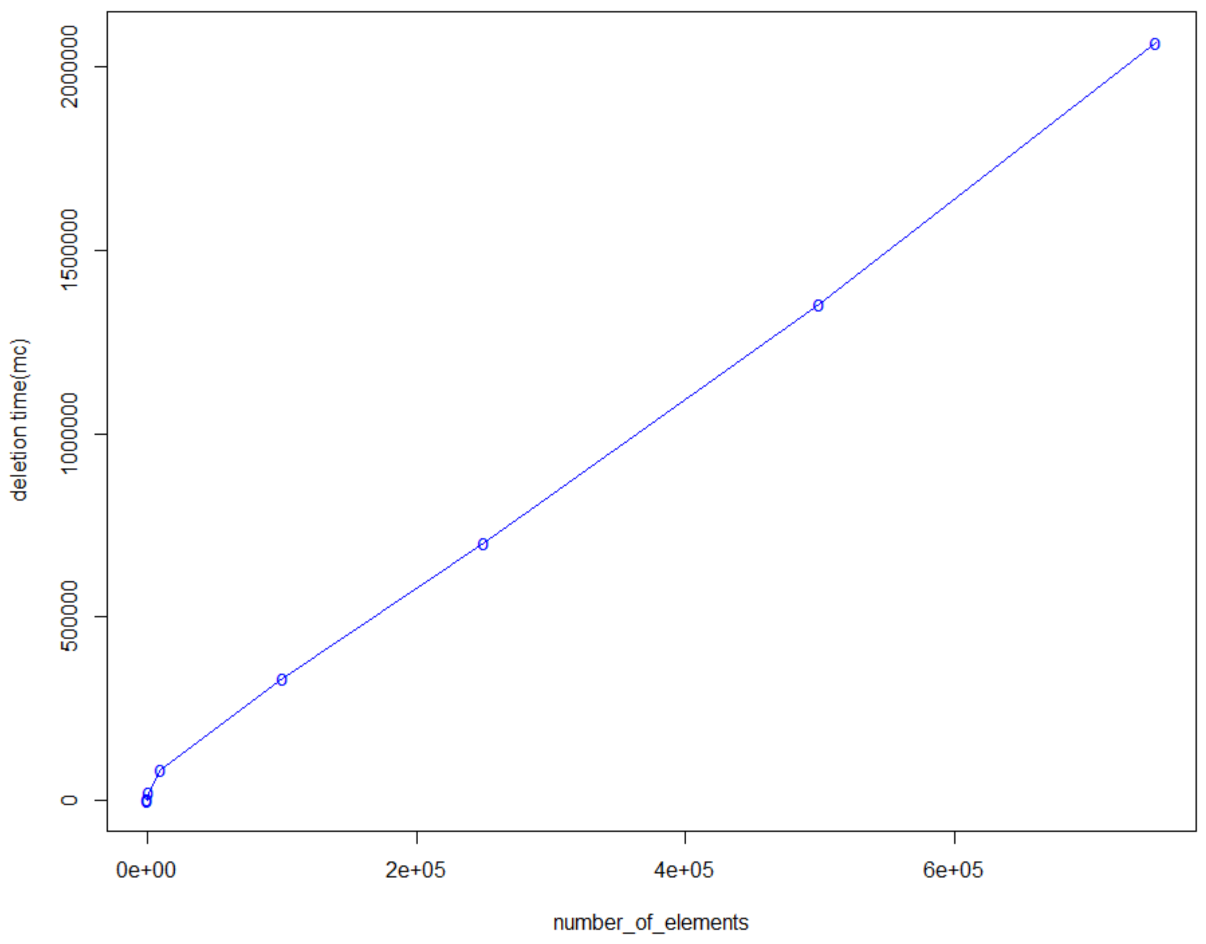
N/trial	0	1	2	3	4	5	6	7	8	9	Average
10	808800	202900	103700	125600	97000	86700	200700	127500	85700	86100	192470
100	2120100	1510200	812400	1326800	904600	821900	8685400	1918600	1141300	922500	2016380
1,000	18439900	17398100	13985700	24180400	70183600	5723400	37658400	11350400	16531000	13890400	22934130
10,000	123649600	72934700	144398800	48881600	49449400	115804800	51190100	133408700	63737200	51793200	85524810
100,000	1590647100	365032400	272110800	161597900	147667800	177242800	145447100	128465600	125416600	208416700	332204480
250,000	1333765600	969343800	882012600	435747100	632397400	699689200	641492500	561554500	413923400	455662700	702558880
500,000	2814888800	1663390700	994556200	900872000	1204469400	1053273100	1095698100	1197863700	1242253600	1360492300	1352775790
750,000	3805820600	1938704900	1682485700	2082841200	2320874300	1711281200	1565801700	1672141200	1871720500	2028135900	2067980720

Graphs:

Insertion Graph



Deletion Graph



Complexity:

From the graph both insertion & deletion times in B-Tree are almost O(log n) and since space complexity of any tree is O(log n) & each node has lists of size O(n) then space complexity is O(n log n).

Search Engine:

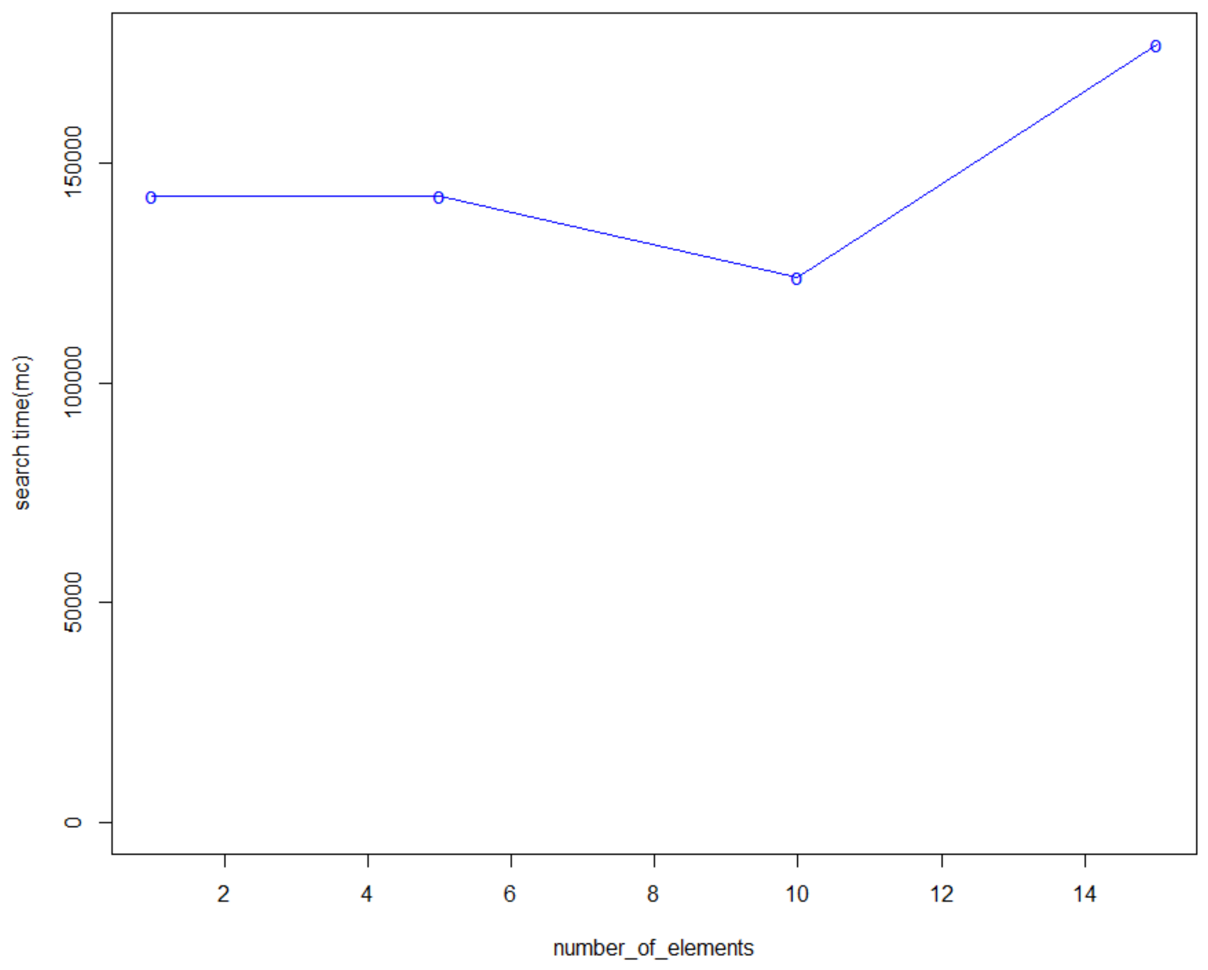
Tables:

Look up multiple words table

trial number	0	1	2	3	4	5	6	7	8	9	Average
1	3E+08	2.98E+08	1.19E+08	1.04E+08	96485200	1.01E+08	96293800	1.04E+08	1.05E+08	1.01E+08	142673970
5	5.68E+08	6.84E+08	4.28E+08	3.99E+08	3.99E+08	4.55E+08	4.03E+08	3.98E+08	5.12E+08	4.17E+08	142673970
10	1.15E+09	7.11E+08	5.63E+08	6.63E+08	6.1E+08	5.49E+08	5.29E+08	5.78E+08	7.48E+08	5.48E+08	124087570
15	1.55E+09	1.45E+09	1.46E+09	1.09E+09	1.74E+09	1.12E+09	1.08E+09	1.29E+09	1.1E+09	9.67E+08	177167100

Graphs:

Look up multiple words Graph



Complexity:

Search Engine searches in each node of the tree which takes O(log n) and then searches in each node’s list of values to find the proper occurrences with O(n) then the total is O(n log n) in terms of space the search engine has only one B-Tree data structure which takes O(n log n) space.