What are the differences between segment trees, interval trees, binary indexed trees and range trees?

Ask Question

149

What are differences between segment trees, interval trees, binary indexed trees and range trees in terms of:

- Key idea/definition
- Applications
- Performance/order in higher dimensions/space consumption

138

Please do not just give definitions.

algorithm tree graph-algorithm interval-tree segment-tree

share improve this question

edited Aug 3 '13 at 3:53

Michael Petrotta
49.7k 12 124 170

asked Jul 4 '13 at 9:04

Aditya
3,408 4 22 43

11 It is not a duplicate, That question is if fenwick trees is generalization of interval tress, and my question is more specific and different. – Aditya Jul 4 '13 at 9:08

6 It has not been answered at stackoverflow.com/questions/2795989/..., the answer there just gives definition. – Aditya Jul 4 '13 at 9:10

11 How is it too broad? "What are some differences between x and y?" is as clear and focused as it gets. This is a very good question. – IVlad Jul 4 '13 at 9:29

13 And there is no good answer for this available anywhere. A good answer will be great for the community – Aditya Jul 4 '13 at 9:30

20 Most of these data structures (except Fenwick trees) are reviewed in this pdf: "Interval, Segment, Range, and Priority Search Trees" (by D. T. Lee). Or you can read it as a chapter from this book: "Handbook of Data Structures and Applications". – Evgeny Kluev Jul 4 '13 at 10:18

Not a greatest answer but still worth reading quora.com/Data-Structures/... – banarun Jul 4 '13 at 10:21

2 I read that already before asking the question, its not really good – Aditya Jul 4 '13 at 10:22

1 If I didn't know about TopCoder et al., this list would look pretty random. Only segment trees and interval trees have the same applications, and there are several divide-and-conquer data structures missing. – David Eisenstat Jul 6 '13 at 13:00

1 The "Interval, Segment, Range, and Priority Search Trees" (by D. T. Lee) was very helpful. – Sangcheol Choi Aug 16 '13 at 9:33

add a comment

2 Answers

active oldest votes

All these data structures are used for solving different problems:

238

- **Segment tree** stores intervals, and optimized for "*which of these intervals contains a given point*" queries.
- **Interval tree** stores intervals as well, but optimized for "*which of these intervals overlap with a given interval*" queries. It can also be used for point queries - similar to segment tree.
- **Range tree** stores points, and optimized for "*which points fall within a given interval*" queries.
- **Binary indexed tree** stores items-count per index, and optimized for "*how many items are there between index m and n*" queries.

+25

Performance / Space consumption for one dimension:

- **Segment tree** - O(n logn) preprocessing time, O(k+logn) query time, O(n logn) space
- **Interval tree** - O(n logn) preprocessing time, O(k+logn) query time, O(n) space
- **Range tree** - O(n logn) preprocessing time, O(k+logn) query time, O(n) space
- **Binary Indexed tree** - O(n logn) preprocessing time, O(logn) query time, O(n) space

(k is the number of reported results).

All data structures can be dynamic, in the sense that the usage scenario includes both data changes and queries:

- **Segment tree** - interval can be added/deleted in O(logn) time (see here)
- **Interval tree** - interval can be added/deleted in O(logn) time
- **Range tree** - new points can be added/deleted in O(logn) time (see here)
- **Binary Indexed tree** - the items-count per index can be increased in O(logn) time

Higher dimensions (d>1):

- **Segment tree** - O(n(logn)^d) preprocessing time, O(k+(logn)^d) query time, O(n(logn)^(d-1)) space
- **Interval tree** - O(n logn) preprocessing time, O(k+(logn)^d) query time, O(n logn) space
- **Range tree** - O(n(logn)^d) preprocessing time, O(k+(logn)^d) query time, O(n(logn)^(d-1))) space
- **Binary Indexed tree** - O(n(logn)^d) preprocessing time, O((logn)^d) query time, O(n(logn)^d) space

share improve this answer

edited Jul 7 '13 at 20:14

answered Jul 6 '13 at 15:49

Lior Kogan
14.3k 3 42 73

11 I really get the impression that segment trees < interval trees from this. Is there any reason to prefer a segment tree? E.g. implementation simplicity? – j_random_hacker Jul 24 '13 at 21:36

5 @j_random_hacker: Segment trees based algorithms have advantages in certain more complex high-dimensional variants of the intervals query. For example, finding which non-axis-parallel line-segments intersect with a 2D window. – Lior Kogan Jul 25 '13 at 16:39

4 Thanks, I'd be interested in any elaboration you could give on that. – j_random_hacker Jul 25 '13 at 23:17

2 @j_random_hacker, segment trees have another interesting use: RMQs (range minimum queries) in O(log N) time where N is the overall interval size. – ars-longa-vita-brevis Feb 26 '14 at 6:48

@LiorKogan Segment trees should only take up O(n) in one dimension if implemented properly. Use an array of size 2*n. – Nicholas Pipitone Jun 21 '16 at 14:36

@NicholasPipitone: This is true when you're using a segment tree for finding sums of a given range. For interval queries, as discussed above, you'll first need to sort the intervals by their endpoints. – Lior KoganJun 26 '16 at 9:23

Maybe a 1D Binary Indexed Tree can be built in O(n)? stackoverflow.com/questions/31068521/… – ThiloJun 13 '17 at 9:30

I think `B+ tree` can absolutely do what `Range` and `binary indexed` tree can do. Am I right? It can also do what interval/segment tree can do but may be not as efficient. – M.kazem Akhgary Feb 7 at 12:33

add a comment

Not that I can add anything to Lior's answer, but it seems like it could do with a good table.

12

One Dimension

`k` is the number of reported results

	Segment	Interval	Range	Indexed
Preprocessing	n logn	n logn	n logn	n logn
Query	k+logn	k+logn	k+logn	logn
Space	n	n	n	n
Insert/Delete	logn	logn	logn	logn

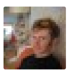
Higher Dimensions

`d > 1`

	Segment	Interval	Range	Indexed
Preprocessing	n(logn)^d	n logn	n(logn)^d	n(logn)^d
Query	k+(logn)^d	k+(logn)^d	k+(logn)^d	(logn)^d
Space	n(logn)^(d-1)	n logn	n(logn)^(d-1))	n(logn)^d

These tables are created in Github Formatted Markdown - see Gist if you want the raw text.

share improve this answer edited Dec 20 '17 at 9:17 answered Jan 9 '16 at 22:07



icc97
5,817 5 36 59

2 What do you mean by reported results ? – Pratik Singhal Feb 1 '16 at 12:23

@ps06756 search algorithms often have a runtime of log(n) where n is the inputsize but can yield results that are linear in n which can't be done in logarithmic time (outputting n numbers in log(n) time is not possible). – oerpli Aug 23 '16 at 12:14