Tournament Tree (Winner Tree) and Binary Heap

Given a team of N players. How many minimum games are required to find second best player?

We can use adversary arguments based on tournament tree (Binary Heap).

Tournament tree is a form of min (max) heap which is a complete binary tree. Every external node represents a player and internal node represents winner. In a tournament tree every internal node contains winner and every leaf node contains one player.

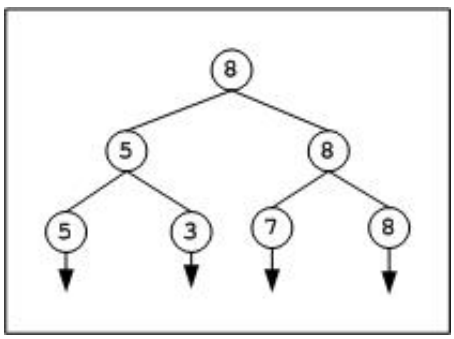
There will be N – 1 internal nodes in a binary tree with N leaf (external) nodes. For details see this post (put n = 2 in equation given in the post).

It is obvious that to select the best player among N players, (N – 1) players to be eliminated, i.e. we need minimum of (N – 1) games (comparisons). Mathematically we can prove it. In a binary tree I = E – 1, where I is number of internal nodes and E is number of external nodes. It means to find maximum or minimum element of an array, we need N – 1 (internal nodes) comparisons.

Second Best Player

The information explored during best player selection can be used to minimize the number of comparisons in tracing the next best players. For example, we can pick second best player in **$(N + \log_2 N - 2)$** comparisons. For details read this comment.

The following diagram displays a tournament tree (*winner tree*) as a max heap. Note that the concept of *loser tree* is different.



The above tree contains 4 leaf nodes that represent players and have 3 levels 0, 1 and 2. Initially 2 games are conducted at level 2, one between 5 and 3 and another one between 7 and 8. In the next move, one more game is conducted between 5 and 8 to conclude the final winner. Overall we need 3 comparisons. For second best player we need to trace the candidates participated with final winner, that leads to 7 as second best.

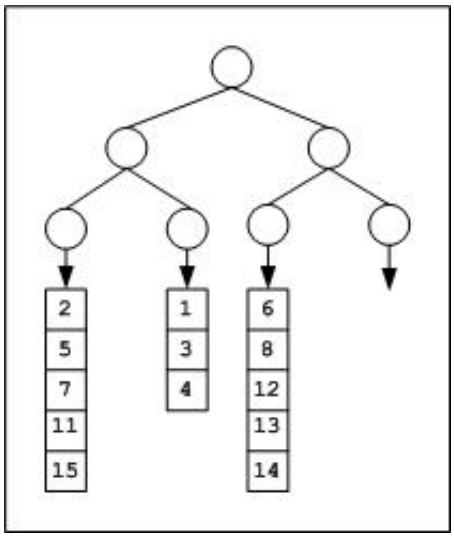
Median of Sorted Arrays

Tournament tree can effectively be used to find median of sorted arrays. Assume, given M sorted arrays of equal size L (for simplicity). We can attach all these sorted arrays to the tournament tree, one array per leaf. We need a tree of height **CEIL ($\log_2$M)** to have atleast M external nodes.

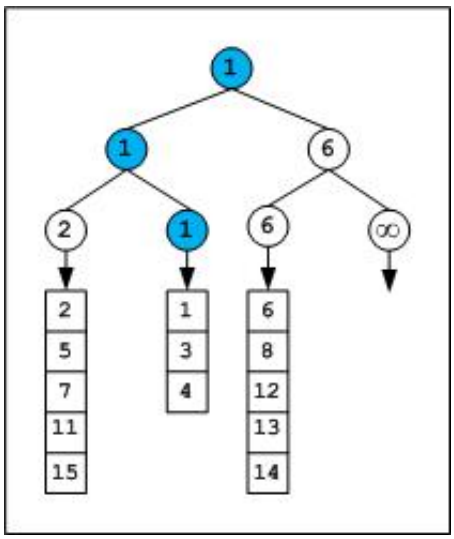
Consider an example. Given 3 (M = 3) sorted integer arrays of maximum size 5 elements.

```
{ 2, 5, 7, 11, 15 } ---- Array1
{1, 3, 4} ---- Array2
{6, 8, 12, 13, 14} ---- Array3
```

What should be the height of tournament tree? We need to construct a tournament tree of height $\log_2 3$.= 1.585 = 2 rounded to next integer. A binary tree of height 2 will have 4 leaves to which we can attach the arrays as shown in the below figure.



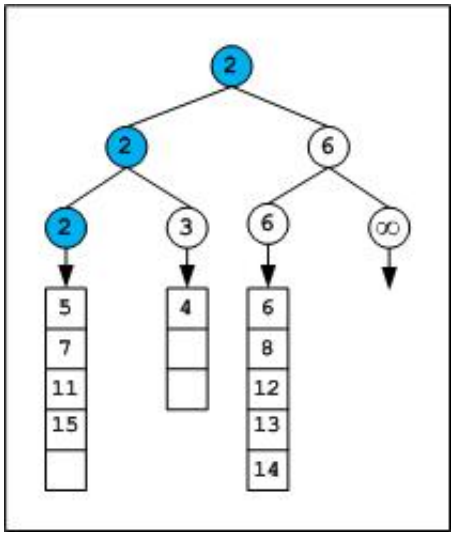
After the first tournament, the tree appears as below,



We can observe that the winner is from Array2. Hence the next element from Array2 will dive-in and games will be played along the winner path of previous tournament.

Note that infinity is used as sentinel element. Based on data being hold in nodes we can select the sentinel character. For example we usually store the pointers in nodes rather than keys, so NULL can serve as sentinel. If any of the array exhausts we will fill the corresponding leaf and upcoming internal nodes with sentinel.

After the second tournament, the tree appears as below,



The next winner is from Array1, so next element of Array1 array which is 5 will dive-in to the next round, and next tournament played along the path of 2.

The tournaments can be continued till we get median element which is (5+3+5)/2 = 7th element. Note that there are even better algorithms for finding median of union of sorted arrays, for details see the related links given below.

In general with M sorted lists of size L_1, L_2 … L_m requires time complexity of $O((L_1 + L_2 + \ldots + L_m) * logM)$ to merge all the arrays, and $O(m*logM)$ time to find median, where m is median position.

Select smallest one million elements from one billion unsorted elements:

As a simple solution, we can sort the billion numbers and select first one million.

On a limited memory system sorting billion elements and picking the first one million seems to be impractical. We can use tournament tree approach. At any time only elements of tree to be in memory.

Split the large array (perhaps stored on disk) into smaller size arrays of size one million each (or even smaller that can be sorted by the machine). Sort these 1000 small size arrays and store them on disk as individual files. Construct a tournament tree which can have atleast 1000 leaf nodes (tree to be of height 10 since $2^9 < 1000 < 2^{10}$, if the individual file size is even smaller we will need more leaf nodes). Every leaf node will have an engine that picks next element from the sorted file stored on disk. We can play the tournament tree game to extract first one million elements.

Total cost = sorting 1000 lists of one million each + tree construction + tournaments

Implementation

We need to build the tree (heap) in bottom-up manner. All the leaf nodes filled first. Start at the left extreme of tree and fill along the breadth (i.e. from 2^{k-1} to $2^k - 1$ where k is depth of tree) and play the game. After practicing with few examples it will be easy to write code. We will have code in an upcoming article.

Related Posts

Link 1, Link 2, Link 3, Link 4, Link 5, Link 6, Link 7.

— by **Venki**. Please write comments if you find anything incorrect, or you want to share more information

about the topic discussed above.

39 Comments Category: Trees

About Venki

Software Engineer

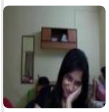
View all posts by Venki →

Related Questions:

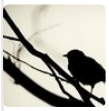
- Data Structure for a single resource reservations
- How to handle duplicates in Binary Search Tree?
- Handshaking Lemma and Interesting Tree Properties
- Advantages of BST over Hash Table
- Given a binary tree, how do you remove all the half nodes?
- K'th Largest Element in BST when modification to BST is not allowed
- Vertex Cover Problem | Set 2 (Dynamic Programming Solution for Tree)
- Check whether a binary tree is a complete tree or not | Set 2 (Recursive Solution)

39 Comments GeeksforGeeks Priyanka Khire

Recommend Share Sort by Newest



Join the discussion…



Holden · 14 days ago

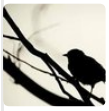
Can anyone please explain why between 5 and 7, 7 is the winner?

since, the author considered: "In the median of sorted arrays case the winner is an element with smaller numerical value."

So, between 5 and 7, I think 5 should be second winner!

what is the procedure for finding second winner?

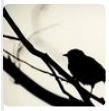
Reply · Share ›



Holden · 14 days ago

"We will have code in an upcoming article."
If there is implementation, could you please provide the link?

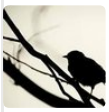
Reply · Share ›



Holden · 14 days ago

Please check links at the end the post:
"Link 1" and "Link 3" are the same, and there is nothing there; only the question.
"Link 6" and "Link 4" as well; there is nothing there!

Reply · Share ›



Holden · 14 days ago

This link is broken:

"For example, we can pick second best player in (N + log2N – 2) comparisons. For details read this comment."

Reply · Share ›

Aditya Goel · 7 months ago

ANY player defeated by the best player can be the second-best, not always the one who lasts till the end. So, after getting the best player, we must carry out another tournament between the players defeated directly by the best to find the second-best player. Therefore competition between 5 and 7 will decide 2nd best player.

1 Reply · Share ›

pink · 8 months ago

For finding the second best player, we need to repeat the entire process among all the players that participated with final winner and find the winner among them. i.e. we will have competition among 5 and 7 and winner of the competition will be final winner.

Reply · Share ›

prashant jha · a year ago

for median of sorted array using tournament tree
http://ideone.com/aqUeRy

Reply · Share ›

santhosh ➚ prashant jha · a year ago

hey prasant

can u explain your code why are you iterating unto 6 and level started as 2 means log n where n is number of arrays as height ? what is fun doing please elaborate

1 Reply · Share ›

prashant jha · a year ago

here is my implementation for median of sorted array using tournament tree

```
#include<iostream>
#define infinity 999
using namespace std;
```

```
struct tnode
{
tnode* lchild;
int data;
tnode* rchild;
int *aptr;
int ind;
tnode()
{
lchild=NULL;
aptr=NULL;
ind=0;
rchild=NULL;
```

see more

Reply · Share ›

danny · a year ago

For second best player we need to trace the candidates participated with final winner, that leads to 7 as second best.

It must be 5 instead of 7 please admin correct me if I am wrong, This is the last line to find the second best player.

Reply · Share ›

danny · a year ago

For second best player we need to trace the candidates participated with final winner, that leads to 7 as second best.

It must be 5 instead of 7 please admin correct me if I am wrong, This is the last line to find the second best player.

3 Reply · Share ›

Avinash Abhi · 2 years ago

(n-1) comparison for finding the winning element, so the elements that lost to the winning element in the winning path will be [log(n)-1].
so for finding the next winning element,
there will be [{log(n)-1} -1] comparison along the winning path.

we see the minimum comparison comes out to be [n+ log(n) - 3].

so how the minimum comparison for second winning element is calculated to be [n+log(n)-2]?

1 Reply · Share ›

Audi ➚ Avinash Abhi · 6 months ago

the elements that lost to the winning element in the winning path will be log(n)

log(n)

5 • Reply • Share ›

Holden → Audi • 14 days ago

Can you explain why between 5 and 7, 7 is the winner?

since, the author considered: "In the median of sorted arrays case the winner is an element with smaller numerical value."

I think 5 should be second winner!

what is the procedure for finding second winner?

• Reply • Share ›

Pranjal Gupta • 2 years ago

awesome article..

• Reply • Share ›

geeky • 3 years ago

does it means we first make a max heap and then backtrack to find the initial competitor of the 1st best player....it would be great if you post the code..

• Reply • Share ›

Nitin • 4 years ago

can u provide me basic code for tournament trees

• Reply • Share ›

levis • 4 years ago

@venki nice post !gives more clarity to the problem !!

• Reply • Share ›

vinit • 4 years ago

nice :)

• Reply • Share ›

Algoseekar • 4 years ago

@venki plz post code for above

• Reply • Share ›

Venki → Algoseekar • 4 years ago

@Algoseekar, sure. I need some time.

• Reply • Share ›

sesha → Venki • 3 years ago

@venki,
I have implemented Tournament Tree for Merging K sorted arrays.
http://yourbitsandbytes.com/vi...

Hope this helps.

2 ^ | v • Reply • Share ›

Holden ➚ sesha • 14 days ago

Great! Thank you!

Can you explain why between 5 and 7, 7 is the winner?

since, we consider: "In the median of sorted arrays case the winner is an element with smaller numerical value."

I think 5 should be second winner.

^ | v • Reply • Share ›

slimshady • 4 years ago

"The above tree contains 4 leaf nodes that represent players and have 3 levels 0, 1 and 2. Initially 2 games are conducted at level 2, one between 5 and 3 and another one between 7 and 8. In the next move, one more game is conducted between 5 and 8 to conclude the final winner. Overall we need 3 comparisons. For second best player we need to trace the candidates participated with final winner, that leads to 7 as second best."

above explanation concludes 7 as second best.. is this right ?? or it should be 5 ??

1 ^ | v • Reply • Share ›

Venki ➚ slimshady • 4 years ago

@slimshady, why confusion? The candidates participated with the winner are 5 and 7. So, 7 is next higher number (next best player).

^ | v • Reply • Share ›

Holden ➚ Venki • 14 days ago

Based on this fact which you said earlier in comments:

"In the median of sorted arrays case the winner is an element with smaller numerical value."

Between, 5 and 7, 5 should be the winner!

^ | v • Reply • Share ›

slimshady ➚ Venki • 4 years ago

@venki: thanks for clarifying. I was under impression since 5 played with 8 at an higher level(say semi-finals) it made him runner up n next best player after 8..

^ | v • Reply • Share ›

slimshady ➚ slimshady • 4 years ago

@venki: sorry for my last comment.. i meant since 5 played with 8 at an higher level(say finals) it made him runner up n

with 8 at an higher level(say finals) it made him runner up n next best player after 8..

^ | v • Reply • Share ›

bbqabbq ➚ slimshady • 3 years ago

the problem description of "second best player" is vague, since 5 and 7 are all losers, what is the second best?

```
/* Paste your code here (You may delete these lines if
```

1 ^ | v • Reply • Share ›

Holden ➚ bbqabbq • 14 days ago

I have the same question!

^ | v • Reply • Share ›

abc • 4 years ago

In the median of sorted array solution: how is "In general with M sorted lists of size L1, L2 … Lm requires time complexity of O(max(L1, L2 … Lm) * logM)." calculated? Thanks!

1 ^ | v • Reply • Share ›

Venki ➚ abc • 4 years ago

@abc, thanks for pointing. I have updated the post.

1 ^ | v • Reply • Share ›

slimshady • 4 years ago

can someone explain
"The tournaments can be continued till we get median element which is (5+3+5)/2 = 7th element."

how can we use tournament tree to calculate median ?

thanks

^ | v • Reply • Share ›

abc ➚ slimshady • 4 years ago

there are 3 arrays of sizes 5,3, and 5 respectively. So total number of elements would be 5+3+5 = 13. the median of 13 elements would be 7th element when arranged in sorted order.
So we play tournament (loser tree) to find the the 7th min element

1 ^ | v • Reply • Share ›

Venki ➚ abc • 4 years ago

@abc, The concept of *loser tree* is different, same thing I emphasized in the post. We covered only winner tree here. In the median of sorted arrays case the winner is an element with smaller numerical

value.

In otherwords, elements with smaller values get high priority in the game, and still the winner moves up.

In a winner tree whatever may be the winner, whether smaller value number wins or larger value numbers wins, only the winner moves up. In case of loser tree, both the loser and winner moves up in the tree. We will cover it later.

1 · Reply · Share ›

Holden ➚ Venki · 14 days ago

"We will cover it later." If there is a post under loser tree, could you please give its link? thanks

· Reply · Share ›

abc ➚ Venki · 4 years ago

@Venki: Ahh ok! Will wait for the post

1 · Reply · Share ›

slimshady · 4 years ago

very interesting .. looking forward for the code :)

1 · Reply · Share ›

Holden ➚ slimshady · 14 days ago

Me also! after 4 years still looking forward to seeing the code! :))