Wenlan Tian

COP 3530

Section 1087, MAEB 211

10/14/2014

Homework 4

"On my honor, I have neither given nor received unauthorized aid in doing this assignment.”

Summary of learning experience:

The easiest part of the task is to understand how the array should be changed every time. Compare the value in each two array index, and only the winner will be recorded to a new array.

The most difficult parts of the task: how to calculate the size of the new added array because of even number or odd number.

How well I think I achieved them: I finished the pseudocode, but it may not be a very smart way. I hope I can get a more easy way to implement it.

Wenlan Tian

COP 3530

Section 1087, MAEB 211

10/14/2014

Homework 4

Homework 4:

Figure out an algorithm for generating a packed array representation of a tournament tree.

This is a psedocode to implement tournament tree using packed array. The basic idea is using two arrays, the old array and a new array. The new array is the half of the old array containing the winner between each two index, plus the old array. Each loop, the new added array should be around half of the previous added array. So at last, it will become 1, and exit the loop.

Assume the original size of number is n;

int \* make TT(int\* score\_start, int\* score\_end){

int array\_size = n;

int array[n];

int new\_add\_array = (n-1)/2+1;

while(new\_add\_array != 1){

int arr\_new[new\_add\_array+ array\_size]

//compare every two array index, and only the winner will go to the new array

for(int i =0; i< new\_add\_array; i++){

arr\_new[i] = (arr[2\*i] > arr[2\*i+1] ? arr[2\*i] : arr[2\*i+1])

}

//copy the old array to the remaining old array

for(int i = new\_add\_array; i< new\_add\_array + array\_size; i++){

arr\_new[i] = arr[i-new\_add\_array];

}

arr = arr\_new; // copy the new array to the old array

arr\_size = array\_size + new\_add\_array;

new\_add\_array = (new\_add\_array-1)/2+1;

}

}