**Tournament Sort**

**DEVELOPMENT**

- Alexander Stepanov, Russian computer programmer, who is known best for his primary design and implementation on the c++ library.

- Aaron Kershenbaum, Engineer, who is known best for his book titled “Telecommunications Network Design Algorithm.”

- It was developed in the year 2002.

**ORIGINAL CODE**

* It is a generalization of binomial trees of Brown and Vuillemin.
* It is based on selection sort but it uses a priority queue.
* It is a variation of a heapsort and sometimes heapsort can be called tournament sort.
* The name is tournament sort because it is similar to a single elimination game.

**ALGORITHM**

//definitions and declarations

#define INF 9999999

#define SIZE 5

typedef struct{

int elem[SIZE\*2-1];

int lastNdx;

}Heap;

int data[SIZE\*2-1];

Heap tournament\_tree;

void tournament\_sort(int data[]){

int ctr,x;

Heap tournament\_tree = create\_tree(data);

for(ct r= 0; ctr < SIZE; ctr++){

for(x = (tournament\_tree.lastNdx) / 2 - 1; x > -1 ; x--){

play(&tournament\_tree, x);

}

}

}

Heap create\_tree(int data[]) {

Heap tournament\_tree;

tournament\_tree.lastNdx = size\*2-2;

int x,y;

for(x=0 ; x<SIZE-1 ; x++){

tournament\_tree.elem[x] = INF;

}

for(x, y=0 ; y<SIZE\*2-1; x++, y++){

tournament\_tree-.elem[x] = data[y];

}

return tournament\_tree;

}

void play(Heap \*L, int parent){

int left\_child = 2\*parent + 1;

int right\_child = 2\*parent + 2;

int smallest = left\_child;

if(right\_child <= L->lastNdx && L->elem[right\_child]< L->elem[smallest]){

smallest = right\_child;

}

L->elem[parent] = L->elem[smallest];

}

void heapify\_subtree(Heap \*L, int parent){

int smallest = parent;

int left\_child = 2\*parent + 1;

int right\_child = 2\*parent + 2;

if(left\_child <= L->lastNdx && L->elem[left\_child] < L->elem[smallest]){

smallest = left\_child;

}

if(right\_child <= L->lastNdx && L->elem[right\_child]< L->elem[smallest]){

smallest = right\_child;

}

if(smallest != parent){

L->elem[smallest] = INF;

heapify\_subtree(L, smallest);

}

}