**QUEUE.H**

*// COS30008, Problem Set 7, 2020*

#pragma once

#include "List.h"

#include <stdexcept>

template<class T>

class Queue

{

private:

List<T> fElements;

public:

typedef typename List<T>::Iterator Iterator;

bool isEmpty() const

{

return fElements.isEmpty();

}

int size() const

{

return fElements.size();

}

void enqueue( const T& aElement )

{

fElements.push\_back( aElement );

}

void dequeue()

{

if ( isEmpty() )

throw std::underflow\_error( "Queue empty." );

fElements.remove( fElements[0]);

}

const T& top() const

{

if ( isEmpty() )

throw std::underflow\_error( "Queue empty." );

return fElements[0];

}

*// stack indexer*

const T& operator[]( size\_t aIndex ) const

{

assert(aIndex < fElements);

return fElements[aIndex];

}

Iterator begin() const *// return a forward iterator*

{

return fElements.begin();

}

Iterator end() const *// return a forward end iterator*

{

return fElements.end();

}

Iterator rbegin() const *// return a backwards iterator*

{

return fElements.rbegin();

}

Iterator rend() const *// return a backwards end iterator*

{

return fElements.rend();

}

};

**NTREE.H**

#if P>=2

*// depth-first traversal*

void traverseDepthFirst( const TreeVisitor<T>& aVisitor ) const

{

if ( !isEmpty() )

{

aVisitor.preVisit( key() );

for ( unsigned int i = 0; i < N; i++ )

{

fNodes[i]->traverseDepthFirst( aVisitor );

}

aVisitor.postVisit( key() );

}

}

#endif

#if P>=3

*// breadth-first traversal*

void traverseBreadthFirst(const TreeVisitor<T>& aVisitor) const

{

Queue<const NTree<T, N>\*> lQueue;

if (!isEmpty())

{

lQueue.enqueue(this);

while (!lQueue.isEmpty())

{

const NTree<T, N>\* lTree = lQueue.top();

lQueue.dequeue();

aVisitor.visit(lTree->key());

for (unsigned int i = 0; i < N; i++)

{

if (!lTree->fNodes[i]->isEmpty())

{

lQueue.enqueue(lTree->fNodes[i]);

}

}

}

}

}

#endif

A screenshot of a cell phone

Description automatically generated

A black sign with white text

Description automatically generatedA black sign with white text

Description automatically generated