

Name : Swaminathan Navinashok

Roll No: 2019115126

1. Create a Book Index which consists of Key terms, Description and Page Number (Use Splay tree for implementation). Perform the following operations:
 - a. Insert into the Book index with appropriate key term, description and page number if a new term comes.
 - b. Delete the entry from the book index if the term is no longer needed.
 - c. If key term is given as input, corresponding description has to be retrieved and the key term should splay to the top of the tree.
 - d. Find the key terms which are frequently searched. Display the number of times each key term has been searched in descending order.
 - e. Suitable display operations for every operation stated above.
- f. Display of all key terms if a partial match occurs for 3 letters – Once the user chooses the appropriate term from the list of key terms displayed – corresponding description has to be displayed

“Main.cpp”

```
#include <iostream>
#include <cstdlib>
#include<string>
#include<vector>
```

```

#include<algorithm>
using namespace std;
#include "classes.h"

bool func(Node<book_rec*>*& book1,Node<book_rec*>*& book2)//v.v.imp must pass
parameters by reference
{
    return ((book1->element).freq)>((book2->element).freq);
}

vector<Node<book_rec*>> sort_by_freq(Splay<book_rec>& at,Node<book_rec*>*& root)
{
    vector<Node<book_rec*>> arr={};
    at.tree_to_vect(root,arr);
    sort(arr.begin(),arr.end(),func);
    return arr;
    //member function pointers complex
}

int main() {
    Splay<book_rec> at;
    Node<book_rec*>* root = NULL;
    vector<int> output;
}

```

```

vector<Node<book_rec*>> vec;
vector<Node<book_rec*>> vec2;

string key;

int input;
int option=0;
int type;
book_rec temp;
book_rec temp2;
string guess;

do {

cout<<"\n\n\n\n"
<<"\n\n 1. Insert into the Book index with appropriate key term, description and page
number if a new term comes."
<<"\n\n 2. Delete the entry from the book index if the term is no longer needed."
<<"\n\n 3. If key term is given as input, corresponding description has to be retrieved
and the key term should splay to the top of the tree."
<<"\n\n 4. Find the key terms which are frequently searched. Display the number of
times each key term has been searched in descending order."
<<"\n\n 5. Suitable display operations for every operation stated above."
<<"\n\n 6. Display of all key terms if a partial match occurs for 3 letters - Once the user
chooses the appropriate term from the list of key terms displayed - corresponding
description has to be displayed."
<<"\n\n 7. Exit"
<<"\n\n\n\n      Enter option      ";
fflush(stdout);

cin >> option;

```

```
switch (option) {  
  
    case 1:  
  
        cout<<"\n\n\n          Enter book details  ";  
        fflush(stdout);  
        cin>>temp;  
  
        root = at.Insert(temp, root);  
  
        break;  
  
    case 2:  
        cout<<"\n\n\n    enter key(book name) to delete  :  ";cin>>key;  
  
        temp2.key_term=key;  
        root=at.Search(temp2,root);  
        cout<<root->element;  
        root=at.Delete(temp2,root);  
        cout<<"\n\n      deletion of this node was sucessful      ";  
  
        break;  
  
    case 3:
```

```
cout<<"\n\n\n enter key(book name) to seach for : ";cin>>key;
```

```
temp2.key_term=key;  
root=at.Search(temp2,root);  
cout<<root->element;  
  
break;
```

case 4 :

```
cout<<"\n\n\n sorted by frequencies:\n\n " ;  
vec=sort_by_freq(at,root);  
  
for(auto i=begin(vec);i!=end(vec);i++)  
{  
    cout<<"\n\n    "<<(*i)->element).key_term<<" has been searched for  
"<<(*i)->element).freq<<" times\n";  
}  
vec={};//empty the vec after use  
break;
```

case 5:

```
at.display(root);  
break;
```

case 6:

```
cout<<"\n\n\n enter a string, partial match occurs if 3 letters match : " ;  
cin>>guess;  
at.search_guess(root,guess,vec2);  
cout<<"\n\n\n partial matched keys \n\n " ;  
for(auto i=begin(vec2);i!=end(vec2);i++)
```

```

{
    cout<<"\n\n      "<<(*i)->element).key_term;
}

cout<<"\n\n choose key ";
cin>>key;

temp2.key_term=key;

root=at.Search(temp2,root);

cout<<root->element;

vec2={};//empty the vector after use;

break;

```

case 7:

```

cout<<"\n\n";
at.makeEmpty(root);
cout<<"\n\n\n\n";
break;

```

default:

```

cout << "\n\n Enter Proper Option number   " << endl;break;
}

} while (option != 7);

cout<<"\n\n\n          exited  \n\n\n\n\n\n";
return 0;
}
```

“Classes.h”

```
#ifndef CLASSES_H
#define CLASSES_H
#include <iostream>
#include <cstdlib>
#include<string>
#include<vector>
#include<algorithm>
using namespace std;
template <class T>
class Splay;
template <class T>
class Node;

class book_rec
{
public:
    std::string key_term;
    std::string description;
    int pages_num;
    int freq;

    book_rec()
```

```

{

key_term="";
description="";
pages_num=0;
freq=0;

}

book_rec(const book_rec& br)

{
key_term=br.key_term;
description=br.description;
pages_num=br.pages_num;
freq=br.freq;

}

book_rec& operator=(book_rec& br);

friend istream& operator>>(istream& in,book_rec&br)

{
cout<<"\n\n\n\n      Book Name (Key Term) : ";
in.ignore(256, '\n'); //to take care of trailing /n in cin>> (at menu when we choose
option)

getline(in>>ws,br.key_term); //ws takes care of trailing spaces

cout<<"\n\n      Book description : ";
//in.ignore(256, '\n');//no need

getline(in>>ws,br.description);

```

```

        cout<<"\n\n      number of pages : ";

        in>>br.pages_num;

        return in;
    }

friend ostream& operator<<(ostream& out, book_rec&br)
{
    cout<<"\n\n\n      Book Name (Key Term) : "<<br.key_term
        <<"\n\n      Book description : "<<br.description
        <<"\n\n      number of pages : "<<br.pages_num
        <<"\n\n      frequency it was searched for : "
        "<<br.freq;

    return out;
}

bool operator==(book_rec& br)
{
    return (key_term==br.key_term);
}

bool operator<(book_rec& br)//dont make these const
{
    return (key_term<br.key_term);
}

bool operator>(book_rec& br)
{

```

```
    return (key_term>br.key_term);

}

bool operator>=(book_rec& br)
{
    return (operator>(br)||operator==(br));

}

bool operator<=(book_rec& br)
{
    return (operator<(br)||operator==(br));

}

bool operator!=(book_rec& br)
{
    return !(operator==(br));
}

};
```

```
template <class T>
class Node
{
```

```
public:  
    T element;  
    struct Node<T>* left;  
    struct Node<T>* right;  
};  
  
template<class T>  
class Splay  
{  
public:  
  
    Node<T>* New_Node(T key);  
  
    Node<T>* right_rotate(Node<T>* k2);  
  
    Node<T>* left_rotate(Node<T>* k2);  
  
    void display(Node<T>* );  
  
    void inorder(Node<T>* );  
  
    void preorder(Node<T>* );  
  
    Node<T>* splay(T key, Node<T>* root);  
  
    Node<T>* Insert(T key, Node<T>* root);
```

```
void postorder(Node<T>* );

void makeEmpty(Node<T>* );

Node<T>* Delete(T& key, Node<T>* root);

Node<T>* Search(T key, Node<T>* root);

void search_guess(Node<T>*,std::string,std::vector<Node<T>*>&);

void tree_to_vect(Node<T>* ,std::vector<Node<T>*>& );

};

#endif
```

“Classes.cpp”

```
#include <iostream>
#include <cstdlib>
#include<string>
#include<vector>
#include<algorithm>
using namespace std;

book_rec& book_rec::operator=(book_rec& br)

{
    key_term=br.key_term;
    description=br.description;
    pages_num=br.pages_num;
    freq=br.freq;
}

template<typename T>
void Splay<T>::inorder(Node<T>* t)
{
    if(t == NULL)
        return;
    inorder(t->left);
    cout<<"    "<<(t->element).key_term;
    inorder(t->right);
```

```
}
```

```
template <typename T>
void Splay<T>::preorder(Node<T>* p)
{
    if (p!=NULL)
    {
        cout<<"    "<<(p->element).key_term;
        preorder(p->left);
        preorder(p->right);
    }
}
```

```
template <typename T>
void Splay<T>::postorder(Node<T>* p)
{
    if (p!=NULL)
    {
        postorder(p->left);
        postorder(p->right);
        cout<<"    "<<(p->element).key_term;
    }
}
```

```
template <typename T>
```

```

void Splay<T>::display(Node<T>* root)
{
    cout<<"\n\n\n\n    inorder traversal :\n\n\n      ";
    inorder(root);

    cout<<"\n\n\n\n    preorder traversal :\n\n\n      ";
    preorder(root);

    cout<<"\n\n\n\n    postorder traversal :\n\n\n      ";
    postorder(root);

    cout << endl;
}

```

```

template<class T>

inline Node<T>* Splay<T>::left_rotate(Node<T>* k2)
{
    Node<T>* k1 = k2->left;

    k2->left = k1->right;

    k1->right = k2;

    return k1;
}

```

```

template<class T>

inline Node<T>* Splay<T>::right_rotate(Node<T>* k2)
{
    Node<T>* k1 = k2->right;

    k2->right = k1->left;

    k1->left = k2;

    return k1;
}

```

```

template<class T>

Node<T>* Splay<T>::splay(T key, Node<T>* root)

{
    if(!root)
        return NULL;

    Node<T> header;
    header.left = header.right = NULL;
    Node<T>* LeftTreeMax = &header;
    Node<T>* RightTreeMin = &header;

    while(1)
    {
        if(key < root->element)
        {
            if(!root->left)
                break;
            if(key < root->left->element)
            {
                root = left_rotate(root);
                if(!root->left)
                    break;
            }
            /* Link to R Tree */
            RightTreeMin->left = root;
            RightTreeMin = RightTreeMin->left;
            root = root->left;
            RightTreeMin->left = NULL;
        }
    }
}

```

```

    }

else if(key > root->element)

{

    if(!root->right)

        break;

    if(key > root->right->element)

    {

        root = right_rotate(root);

        if(!root->right)

            break;

    }

    LeftTreeMax->right = root;

    LeftTreeMax = LeftTreeMax->right;

    root = root->right;

    LeftTreeMax->right = NULL;

}

else

    break;

}

LeftTreeMax->right = root->left;

RightTreeMin->left = root->right;

root->left = header.right;

root->right = header.left;

return root;

}

```

```

template<class T>

Node<T>* Splay<T>::New_Node(T key)

{
    Node<T>* p_node = new Node<T>;
    if(!p_node)
    {
        fprintf(stderr, "Out of memory!\n");
        exit(1);
    }
    p_node->element = key;
    p_node->left = p_node->right = NULL;
    return p_node;
}

template <class T>

void Splay<T>::makeEmpty(Node<T>* node)

{
    if (node == NULL) return;

    makeEmpty(node->left);
    makeEmpty(node->right);

    cout << "\n\n      Deleting node: " << (node->element).key_term;
    delete node;
}

template<class T>
```

```

Node<T>* Splay<T>::Insert(T key, Node<T>* root)
{
    static Node<T>* p_node = NULL;
    if(!p_node)
        p_node = New_Node(key);
    else
        p_node->element = key;
    if(!root)
    {
        root = p_node;
        p_node = NULL;
        return root;
    }
    root = splay(key, root);

    if(key < root->element)
    {
        p_node->left = root->left;
        p_node->right = root;
        root->left = NULL;
        root = p_node;
    }
    else if(key > root->element)
    {
        p_node->right = root->right;
        p_node->left = root;
        root->right = NULL;
        root = p_node;
    }
}

```

```

        else
            return root;
        p_node = NULL;
        return root;
    }

template<class T>
Node<T>* Splay<T>::Delete(T& key, Node<T>* root)
{
    Node<T>* temp;
    if(!root)
        return NULL;
    root = splay(key, root);
    if(key != root->element)
        return root;
    else
    {
        if(!root->left)
        {
            temp = root;
            root = root->right;
        }
        else
        {
            temp = root;
            root = splay(key, root->left);
        }
    }
}

```

```

        root->right = temp->right;
    }

    free(temp);

    return root;
}

}

template<class T>

Node<T>* Splay<T>::Search(T key, Node<T>* root)

{
    Node<T>* x=splay(key, root);

    (x->element).freq++;

    return x;
}

template <typename T>

void Splay<T>::tree_to_vect(Node<T>* node,vector<Node<T>*>& temp)

{

    if(node==NULL) return;

    temp.push_back(node);

    tree_to_vect(node->right,temp);

    tree_to_vect(node->left,temp);

}

```

```

template <typename T>

    void Splay<T>::search_guess(Node<T>* node,string guess,vector<Node<T>*>&
temp)

{
    if(node==NULL) return;int count=0;

    string str;
    str=(node->element).key_term;

    bool flag=false;

    for(int i=0;i<=str.length()-3;i++)
    {

        for(int j=0;j<=guess.length()-3;j++)
        {

            count=0;
            for(int k=0;k<3;k++)
            {

                if(guess[j+k]==str[i+k])
                    count++;

            }

            if(count==3)
                {temp.push_back(node);flag=true;break;}
        }

        if(flag) break;
    }
}

```

```
    search_guess(node->right,guess,temp);
    search_guess(node->left,guess,temp);

}

template class Splay<book_rec>;
```

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp classes.h classes.cpp

```

267     bool flag=false;
268
269     for(int i=0;i<=str.length()-3;i++)
270     {
271
272
273
274         for(int j=0;j<=guess.length()-3;j++)
275         {
276             count=0;
277             for(int k=0;k<3;k++)
278             {
279                 if(guess[j+k]==str[i+k])
280                     | count++;
281             }
282             if(count==3)
283                 | {temp.push_back(node);flag=true;break;}
284         }
285         if(flag) break;
286
287
288
289
290
291
292
293     search_guess(node->right,guess,temp);
294     search_guess(node->left,guess,temp);
295
296
297
298
299     template class Splay<book_rec>;

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp classes.h classes.cpp

```

258     template <typename T>
259     void Splay<T>::search_guess(Node<T>* node,string guess,vector<Node<T>>& temp)
260     {
261
262         if(node==NULL) return;int count=0;
263         string str;
264         str=(node->element).key_term;
265
266
267
268
269
270         bool flag=false;
271
272         for(int i=0;i<=str.length()-3;i++)
273         {
274
275             for(int j=0;j<=guess.length()-3;j++)
276             {
277                 count=0;
278                 for(int k=0;k<3;k++)
279                 {
280                     if(guess[j+k]==str[i+k])
281                         | count++;
282                 }
283                 if(count==3)
284                     | {temp.push_back(node);flag=true;break;}
285             }
286             if(flag) break;
287
288
289
290
291
292
293
294
295
296
297
298
299

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

232     return root;
233 }
234 }
235
236 template<class T>
237 Node<T>* Splay<T>::Search(T key, Node<T>* root)
238 {
239     Node<T>* xsplay(key, root);
240     (x->element).freq++;
241     return x;
242 }
243
244
245 template <typename T>
246 void Splay<T>::tree_to_vect(Node<T>* node,vector<Node<T>*>& temp)
247 {
248
249     if(node==NULL) return;
250     temp.push_back(node);
251     tree_to_vect(node->right,temp);
252     tree_to_vect(node->left,temp);
253
254 }
255
256
257
258
259 template <typename T>
260 void Splay<T>::search_guess(Node<T>* node,string guess,vector<Node<T>*>& temp)
261 {
262     if(node==NULL) return;int count=0;
263     string str;
264     str=(node->element).key_term;
265 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

207
208     template<class T>
209     Node<T>* Splay<T>::Delete(T& key, Node<T>* root)
210 {
211     Node<T>* temp;
212     if(!root)
213         return NULL;
214     root = splay(key, root);
215     if(key != root->element)
216         return root;
217     else
218     {
219         if(!root->left)
220         {
221             temp = root;
222             root = root->right;
223         }
224         else
225         {
226             temp = root;
227
228             root = splay(key, root->left);
229             root->right = temp->right;
230
231             free(temp);
232         }
233     }
234 }
235
236 template<class T>
237 Node<T>* Splay<T>::Search(T key, Node<T>* root)
238 {
239     Node<T>* xsplay(key, root);
240     (x->element).freq++;
241 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project: Classes Debug

main.cpp classes.h classes.cpp

```
186     if(key < root->element)
187     {
188         p_node->left = root->left;
189         p_node->right = root;
190         root->left = NULL;
191         root = p_node;
192     }
193     else if(key > root->element)
194     {
195         p_node->right = root->right;
196         p_node->left = root;
197         root->right = NULL;
198         root = p_node;
199     }
200     else
201     {
202         return root;
203     }
204 }
205
206
207
208 template<class T>
209 Node<T>* Splay<T>::Delete(T& key, Node<T>* root)
210 {
211     Node<T>* temp;
212     if(!root)
213     {
214         return NULL;
215     }
216     root = splay(key, root);
217     if(key != root->element)
218     {
219         if(!root->left)

Compiler Resources Compile Log Debug Find Results



Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds



Project3 - [Project3.dev] - Dev-C++ 5.11



File Edit Search View Project Execute Tools AStyle Window Help



Project: Classes Debug



main.cpp classes.h classes.cpp



```
168 delete node;
169 }
170 template<class T>
171 Node<T>* Splay<T>::Insert(T key, Node<T>* root)
172 {
173 static Node<T>* p_node = NULL;
174 if(!p_node)
175 {
176 p_node = New_Node(key);
177 }
178 else
179 {
180 p_node->element = key;
181 }
182 if(!root)
183 {
184 root = p_node;
185 p_node = NULL;
186 return root;
187 }
188 root = splay(key, root);
189
190 if(key < root->element)
191 {
192 p_node->left = root->left;
193 p_node->right = root;
194 root->left = NULL;
195 root = p_node;
196 }
197 else if(key > root->element)
198 {
199 p_node->right = root->right;
200 p_node->left = root;
201 root->right = NULL;
202 root = p_node;
203 }
204 else
205 return root;
206 }
```



Compiler Resources Compile Log Debug Find Results



Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds


```

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

main.cpp classes.h classes.cpp
151     p_node->element = key;
152     p_node->left = p_node->right = NULL;
153     return p_node;
154 }

155
156
157     template <class T>
158     void Splay<T>::makeEmpty(Node<T>* node)
159     {
160         if (node == NULL) return;
161
162         makeEmpty(node->left);
163         makeEmpty(node->right);
164
165         cout << "\n\n      Deleting node:      " << (node->element).key_term;
166     }
167
168     template<class T>
169     Node<T>* Splay<T>::Insert(T key, Node<T>* root)
170     {
171         static Node<T>* p_node = NULL;
172         if(!p_node)
173             p_node = New_Node(key);
174         else
175             p_node->element = key;
176         if(!root)
177         {
178             root = p_node;
179             p_node = NULL;
180             return root;
181         }
182         root = splay(key, root);
183     }
184

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

main.cpp classes.h classes.cpp
124     LeftTreeMax->right = root;
125     LeftTreeMax = LeftTreeMax->right;
126     root = root->right;
127     LeftTreeMax->right = NULL;
128 }
129
130     else
131         break;
132 }
133
134     LeftTreeMax->right = root->left;
135     RightTreeMin->left = root->right;
136     root->left = header->right;
137     root->right = header->left;
138
139     return root;
140 }
141
142     template<class T>
143     Node<T>* Splay<T>::New_Node(T key)
144     {
145         Node<T>* p_node = new Node<T>;
146         if(!p_node)
147         {
148             fprintf(stderr, "Out of memory!\n");
149             exit(1);
150         }
151         p_node->element = key;
152         p_node->left = p_node->right = NULL;
153         return p_node;
154     }
155
156     template <class T>
157

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

main.cpp classes.h classes.cpp
99     {
100         if(!root->left)
101             break;
102         if(key < root->left->element)
103         {
104             root = left_rotate(root);
105             if(!root->left)
106                 break;
107         }
108         /* Link to R Tree */
109         RightTreeMin->left = root;
110         RightTreeMin = RightTreeMin->left;
111         root = root->left;
112         RightTreeMin->left = NULL;
113     }
114     else if(key > root->element)
115     {
116         if(!root->right)
117             break;
118         if(key > root->right->element)
119         {
120             root = right_rotate(root);
121             if(!root->right)
122                 break;
123         }
124         LeftTreeMax->right = root;
125         LeftTreeMax = LeftTreeMax->right;
126         root = root->right;
127         LeftTreeMax->right = NULL;
128     }
129     else
130         break;
131 }
132 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

main.cpp classes.h classes.cpp
84     template<class T>
85     Node<T*>* Splay(T key, Node<T*>* root)
86     {
87         if(!root)
88             return NULL;
89         Node<T> header;
90
91         header.left = header.right = NULL;
92         Node<T*> LeftTreeMax = &header;
93         Node<T*> RightTreeMin = &header;
94
95
96         while(1)
97         {
98             if(key < root->element)
99             {
100                 if(!root->left)
101                     break;
102                 if(key < root->left->element)
103                 {
104                     root = left_rotate(root);
105                     if(!root->left)
106                         break;
107                 }
108                 /* Link to R Tree */
109                 RightTreeMin->left = root;
110                 RightTreeMin = RightTreeMin->left;
111                 root = root->left;
112                 RightTreeMin->left = NULL;
113             }
114             else if(key > root->element)
115             {
116                 if(!root->right)
117                     break;
118             }
119         }
120     }
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

main.cpp classes.h classes.cpp
66 template<class T>
67 inline Node<T>* Splay<T>::left_rotate(Node<T*> k2)
68 {
69     Node<T*> k1 = k2->left;
70     k2->left = k1->right;
71     k1->right = k2;
72     return k1;
73 }
74
75 template<class T>
76 inline Node<T>* Splay<T>::right_rotate(Node<T*> k2)
77 {
78     Node<T*> k1 = k2->right;
79     k2->right = k1->left;
80     k1->left = k2;
81     return k1;
82 }
83
84 template<class T>
85 Node<T>* Splay<T>::splay(T key, Node<T*> root)
86 {
87     if(!root)
88         return NULL;
89     Node<T> header;
90
91     header.left = header.right = NULL;
92     Node<T*> LeftTreeMax = &header;
93     Node<T*> RightTreeMin = &header;
94
95     while(1)
96     {
97         if(key < root->element)
98         {
99
46     {
47         postorder(p->left);
48         postorder(p->right);
49         cout<<" " <<(p->element).key_term;
50     }
51
52
53
54 template <typename T>
55 void Splay<T>::display(Node<T*> root)
56 {
57     cout<<"\n\n\n"    inorder traversal : \n\n\n    ";
58     inorder(root);
59     cout<<"\n\n\n"    preorder traversal : \n\n\n    ";
60     preorder(root);
61     cout<<"\n\n\n"    postorder traversal : \n\n\n    ";
62     postorder(root);
63     cout << endl;
64 }
65
66 template<class T>
67 inline Node<T>* Splay<T>::left_rotate(Node<T*> k2)
68 {
69     Node<T*> k1 = k2->left;
70     k2->left = k1->right;
71     k1->right = k2;
72     return k1;
73 }
74
75 template<class T>
76 inline Node<T>* Splay<T>::right_rotate(Node<T*> k2)
77 {
78     Node<T*> k1 = k2->right;
79     k2->right = k1->left;
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Compiler Resources Compile Log Debug Find Results

Line: 288 Col: 19 Sel: 0 Lines: 299 Length: 5709 Insert Done parsing in 0.063 seconds

08:04 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp classes.h classes.cpp

```

26 L   }
27
28
29
30     template <typename T>
31     void Splay<T>::preorder(Node<T>* p)
32     {
33         if (p!=NULL)
34         {
35             cout<<"<<(p->element).key_term;
36             preorder(p->left);
37             preorder(p->right);
38         }
39
40
41
42     template <typename T>
43     void Splay<T>::postorder(Node<T>* p)
44     {
45         if (p!=NULL)
46         {
47             postorder(p->left);
48             postorder(p->right);
49             cout<<"<<(p->element).key_term;
50         }
51
52
53
54     template <typename T>
55     void Splay<T>::display(Node<T>* root)
56     {
57         cout<<"\n\n\n      inorder traversal : \n\n\n      ";
58         inorder(root);
59         cout<<"\n\n\n      preorder traversal : \n\n\n      ";

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 360 Col: 21 Sel: 0 Lines: 384 Length: 9474 Insert Done parsing in 0.062 seconds

07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug main.cpp classes.h classes.cpp

```

1 #include "classes.h"
2 #include <iostream>
3 #include <cstdlib>
4 #include<string>
5 #include<vector>
6 #include<algorithm>
7 using namespace std;
8
9
10 book_rec& book_rec::operator=(book_rec& br)
11 {
12     key_term=br.key_term;
13     description=br.description;
14     pages_num=br.pages_num;
15     freq=br.freq;
16 }
17
18 template<typename T>
19 void Splay<T>::inorder(Node<T>* t)
20 {
21     if(t == NULL)
22         return;
23     inorder(t->left);
24     cout<<"<<(t->element).key_term;
25     inorder(t->right);
26 }
27
28
29
30 template <typename T>
31 void Splay<T>::preorder(Node<T>* p)
32 {
33     if (p!=NULL)
34     {

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 360 Col: 21 Sel: 0 Lines: 384 Length: 9474 Insert Done parsing in 0.062 seconds

07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

116     public:
117     Node<T>* New_Node(T key);
118     Node<T>* left_rotate(Node<T>* k2);
119     Node<T>* right_rotate(Node<T>* k2);
120     void display(Node<T>* );
121     void inorder(Node<T>* );
122     void preorder(Node<T>* );
123     void postorder(Node<T>* );
124     void makeEmpty(Node<T>* );
125     Node<T>* splay(T key, Node<T>* root);
126     Node<T>* Insert(T key, Node<T>* root);
127     Node<T>* Delete(T& key, Node<T>* root);
128     Node<T>* Search(T key, Node<T>* root);
129     void search_guess(Node<T>*,std::string,std::vector<Node<T>>& );
130     void tree_to_vect(Node<T>*,std::vector<Node<T>>& );
131     void
132     };
133     #endiff
134 
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 31 Col: 13 Sel: 0 Lines: 148 Length: 3442 Insert Done parsing in 0.062 seconds

07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

```

102 L     };
103
104     template <class T>
105     class Node
106     {
107     public:
108     T element;
109     struct Node<T>* left;
110     struct Node<T>* right;
111     };
112
113     template<class T>
114     class Splay
115     {
116     public:
117     Node<T>* New_Node(T key);
118     Node<T>* left_rotate(Node<T>* k2);
119     Node<T>* right_rotate(Node<T>* k2);
120     void display(Node<T>* );
121     void inorder(Node<T>* );
122     void preorder(Node<T>* );
123     void postorder(Node<T>* );
124     void splay(T key, Node<T>* root);
125     Node<T>* Insert(T key, Node<T>* root);
126     void
127     };
128     void
129     Node<T>* search_guess(Node<T>*,std::string,std::vector<Node<T>>& );
130     void
131     tree_to_vect(Node<T>*,std::vector<Node<T>>& );
132     void
133     Node<T>*
134     
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 31 Col: 13 Sel: 0 Lines: 148 Length: 3442 Insert Done parsing in 0.062 seconds

07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp classes.h classes.cpp

```

74
75     bool operator==(book_rec& br)
76     {
77         return (key_term==br.key_term);
78     }
79     bool operator<(book_rec& br)//dont make these const
80     {
81         return (key_term<br.key_term);
82     }
83     bool operator>(book_rec& br)
84     {
85         return (key_term>br.key_term);
86     }
87     bool operator>=(book_rec& br)
88     {
89         return (operator>(br)||operator==(br));
90     }
91     bool operator<=(book_rec& br)
92     {
93         return (operator<(br)||operator==(br));
94     }
95     bool operator!=(book_rec& br)
96     {
97         return !(operator==(br));
98     }
99
100
101 }
102
103 template <class T>
104 class Node
105 {
106 public:
107 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search Line: 31 Col: 13 Sel: 0 Lines: 148 Length: 3442 Insert Done parsing in 0.062 seconds

Windows Taskbar: Type here to search, File Explorer, File History, Task View, Start, Taskbar settings, ENG 07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp classes.h classes.cpp

```

42
43     friend istream& operator>>(istream& in,book_rec&br)
44     {
45         cout<<"\n\n\n      Book Name (Key Term) : ";
46         in.ignore(256, '\n'); //to take care of trailing /n in cin>> (at menu when we choose option)
47
48         getline(in>>ws,br.key_term); //ws takes care of trailing spaces
49
50         cout<<"\n\n      Book description : ";
51         //in.ignore(256, '\n');//no need
52
53         getline(in>>ws,br.description);
54
55         cout<<"\n\n      number of pages : ";
56
57         in>>br.pages_num;
58
59         return in;
60     }
61
62
63     friend ostream& operator<<(ostream& out, book_rec&br)
64     {
65         cout<<"\n\n\n      Book Name (Key Term) : "<<br.key_term
66         <<"\n\n      Book description : "<<br.description
67         <<"\n\n      number of pages : "<<br.pages_num
68         <<"\n\n      frequency it was searched for : "<<br.freq;
69
70     }
71
72
73     bool operator==(book_rec& br)
74     {
75 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search Line: 31 Col: 13 Sel: 0 Lines: 148 Length: 3442 Insert Done parsing in 0.062 seconds

Windows Taskbar: Type here to search, File Explorer, File History, Task View, Start, Taskbar settings, ENG 07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug

main.cpp classes.h classes.cpp

```

27     pages_num=0;
28     freq=0;
29 }
30
31 book_rec(const book_rec& br)
32 {
33     key_term=br.key_term;
34     description=br.description;
35     pages_num=br.pages_num;
36     freq=br.freq;
37 }

38
39 book_rec& operator=(book_rec& br);

40
41
42
43 friend istream& operator>>(istream& in,book_rec&br)
44 {
45     cout<<"\n\n\n\n      Book Name (Key Term) :   ";
46     in.ignore(256, '\n'); //to take care of trailing /n in >> (at menu when we choose option)
47
48     getline(in>>ws,br.key_term); //ws takes care of trailing spaces
49
50     cout<<"\n\n      Book description :   ";
51     //in.ignore(256, '\n');//no need
52
53     getline(in>>ws,br.description);
54
55     cout<<"\n\n      number of pages :   ";
56
57     in>>br.pages_num;
58
59     return in;
60 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 31 Col: 13 Sel: 0 Lines: 148 Length: 3442 Insert Done parsing in 0.062 seconds

07:58 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug

main.cpp classes.h classes.cpp

```

1 #ifndef CLASSES_H
2 #define CLASSES_H
3
4 #include <iostream>
5 #include <cstdlib>
6 #include <string>
7 #include <vector>
8 #include <algorithm>
9 using namespace std;
10 template <class T>
11 class Splay;
12 template <class T>
13 class Node;
14
15 class book_rec
16 {
17 public:
18     std::string key_term;
19     std::string description;
20     int pages_num;
21     int freq;
22
23     book_rec()
24     {
25         key_term="";
26         description="";
27         pages_num=0;
28         freq=0;
29     }
30
31     book_rec(const book_rec& br)
32     {
33         key_term=br.key_term;
34         description=br.description;
35     }
36 }
```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 31 Col: 13 Sel: 0 Lines: 148 Length: 3442 Insert Done parsing in 0.062 seconds

07:58 25-04-2021

The image shows two instances of the Dev-C++ IDE running side-by-side. Both instances have the title "Project3 - [Project3.dev] - Dev-C++ 5.11". The top window displays code for handling user input and searching a tree structure. The bottom window displays code for displaying sorted frequencies and performing partial matches.

Top Window (Code for User Input and Search):

```
118     cin>>guess;
119     at.search_guess(root,guess,vec2);
120     cout<<"\n\n\n    partial matched keys  \n\n\n ";
121     for(auto i=begin(vec2);i!=end(vec2);i++)
122     {
123         cout<<"\n\n        "<<(*i->element).key_term;
124     }
125     cout<<"\n\n    choose key  ";
126     cin>>key;
127     temp2.key_term=key;
128     root=&at.Search(temp2,root);
129     cout<<root->element;
130     vec2={};//empty the vector after use;
131     break;
132
133
134
135
136     case 7:
137     cout<<"\n\n";
138     at.makeEmpty(root);
139     cout<<"\n\n\n\n";
140     break;
141
142
143     default:
144     cout << "\n\n    Enter Proper Option number      " << endl; break;
145   }
146
147 } while (option != 7);
148 cout<<"\n\n\n";
149 return 0;
150 }
```

Bottom Window (Code for Display and Partial Match):

```
103     cout<<"\n\n\n\n    sorted by frequencies:\n\n      ";
104     vec=sort_by_freq(at,root);
105
106
107     for(auto i=begin(vec);i!=end(vec);i++)
108     {
109         cout<<"\n\n        "<<(*i->element).key_term<<    has been searched for    <<(*i->element).fre
110     }
111     vec={};//empty the vec after use
112     break;
113
114
115
116
117     case 5:
118     at.display(root);
119     break;
120
121
122     case 6:
123     cout<<"\n\n\n    enter a string, partial match occurs if 3 letters match  :  ";
124     cin>>guess;
125     at.search_guess(root,guess,vec2);
126     cout<<"\n\n\n    partial matched keys  \n\n\n ";
127     for(auto i=begin(vec2);i!=end(vec2);i++)
128     {
129         cout<<"\n\n        "<<(*i->element).key_term;
130     }
131     cout<<"\n\n    choose key  ";
132     cin>>key;
133     temp2.key_term=key;
134     root=&at.Search(temp2,root);
135     cout<<root->element;
136     vec2={};//empty the vector after use;
137     break;
```

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp classes.h classes.cpp

```

87     break;
88
89     case 3:
90
91         cout<<"\n\n\n      enter key(book name) to search for : "; cin>>key;
92
93         temp2.key_term=key;
94         root=at.Search(temp2,root);
95         cout<<root->element;
96
97         break;
98
99     case 4:
100        cout<<"\n\n\n      sorted by frequencies:\n\n      ";
101        vecsort_by_freq(at,root);
102
103        for(auto i=begin(vec);i!=end(vec);i++)
104        {
105            cout<<"\n      " <<(*i)->element->key_term << " has been searched for " <<(*i)->element->freq;
106        }
107
108        vec={};//empty the vec after use
109
110        break;
111
112    case 5:
113        at.display(root);
114        break;
115
116    case 6:
117        cout<<"\n\n\n      enter a string, partial match occurs if 3 letters match : ";
118        cin>>guess;
119        at.search_guess(root,guess,vec2);

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 6 Col: 21 Sel: 0 Lines: 150 Length: 4193 Insert Done parsing in 0.062 seconds

File Edit Search View Project Execute Tools AStyle Window Help

TIM-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp classes.h classes.cpp

```

64     switch (option) {
65
66     case 1:
67
68         cout<<"\n\n\n      Enter book details ";
69         fflush(stdout);
70         cin>>temp;
71
72         root = at.Insert(temp, root);
73
74         break;
75
76     case 2:
77         cout<<"\n\n\n      enter key(book name) to delete : "; cin>>key;
78
79         temp2.key_term=key;
80         root=at.Search(temp2,root);
81         cout<<root->element;
82         root=at.Delete(temp2,root);
83         cout<<"\n      deletion of this node was sucessful ";
84
85
86         break;
87
88     case 3:
89
90         cout<<"\n\n\n      enter key(book name) to search for : "; cin>>key;
91
92         temp2.key_term=key;

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 6 Col: 21 Sel: 0 Lines: 150 Length: 4193 Insert Done parsing in 0.062 seconds

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

main.cpp classes.h classes.cpp

```

46
47 do {
48
49 cout<<"\n\n\n\n"
50     "1. Insert into the Book index with appropriate key term, description and page number if a new term comes."
51     "2. Delete the entry from the book index if the term is no longer needed."
52     "3. If key term is given as input, corresponding description has to be retrieved and the key term should splay
53     "4. Find the key terms which are frequently searched. Display the number of times each key term has been search
54     "5. Suitable display operations for every operation stated above."
55     "6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate te
56     "7. Exit"
57     <<"\n\n\n\n"      Enter option      ";
58     fflush(stdout);
59     cin >> option;
60
61     //Node n1;
62
63     switch (option) {
64
65     case 1:
66
67         cout<<"\n\n\n"          Enter book details  ";
68         fflush(stdout);
69         cin>>temp;
70
71         root = at.Insert(temp, root);
72
73         break;
74
75     case 2:
76         cout<<"\n\n\n"      enter key(book name) to delete   :    ";cin>>key;
77
78 }

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 6 Col: 21 Sel: 0 Lines: 150 Length: 4193 Insert Done parsing in 0.062 seconds

07:57 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIFF-GCC 4.9.2 64-bit Release

Project Classes Debug

main.cpp classes.h classes.cpp

```

16
17 vector<Node<book_rec*>> sort_by_freq(Splay<book_rec>& at,Node<book_rec*>* root)
18 {
19     vector<Node<book_rec*>> arr={};
20     at.tree_to_vect(root,arr);
21     sort(arr.begin(),arr.end(),func);
22     return arr;
23     //member function pointers complex
24
25 }
26
27
28
29 int main() {
30
31     Splay<book_rec> at;
32     Node<book_rec>* root = NULL;
33     vector<int> output;
34     vector<Node<book_rec*>> vec;
35     vector<Node<book_rec*>> vec2;
36
37     string key;
38
39     int input;
40     int option=0;
41     int type;
42     book_rec temp;
43     book_rec temp2;
44     string guess;
45
46
47 do {
48

```

Compiler Resources Compile Log Debug Find Results

Type here to search

Line: 6 Col: 21 Sel: 0 Lines: 150 Length: 4193 Insert Done parsing in 0.062 seconds

07:57 25-04-2021

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

Project Classes Debug

main.cpp classes.h classes.cpp

```
1 #include <iostream>
2 #include <cstdlib>
3 #include<string>
4 #include<vector>
5 #include<algorithm>
6 using namespace std;
7 #include "classes.h"
8
9
10 bool func(Node<book_rec*>& book1,Node<book_rec*>& book2)//v.v.imp must pass parameters by reference
11 {
12     return ((book1->element).freq)>((book2->element).freq);
13 }
14
15
16
17 vector<Node<book_rec*>> sort_by_freq(Splay<book_rec*>& at,Node<book_rec*>& root)
18 {
19
20     vector<Node<book_rec*>> arr={};
21     at.tree_to_vect(root,arr);
22     sort(arr.begin(),arr.end(),func);
23     return arr;
24 //member function pointers complex
```

Compiler Resources CompileLog Debug Find Results Close

Root Compilation

Compilation results...

- Errors: 0

- Warnings: 0

- Output Filename: D:\dev\24_april\trial_3'\Project3.exe

- Output Size: 2.45010089874268 MiB

- Compilation Time: 0.23s

Type here to search

Line: 7 Col: 21 Sel: 0 Lines: 150 Length: 4193 Insert Done parsing in 0.062 seconds

07:57 25-04-2021

```
D:\dev\24 april\trial 3\Project3.exe
    number of pages      :      1024
    frequency it was searched for      :      3
```

1. Insert into the Book index with appropriate key term, description and page number if a new term.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the term, the program has to display all the terms which have a partial match for the given term.
7. Exit

```
Enter option      7
```

```
Deleting node:      AlgoC
Deleting node:      DsJava
Deleting node:      Algorithms
```

```
exited
```

```
Process exited after 468.6 seconds with return value 0
Press any key to continue . . .
```



Type here to search



D:\dev\24 april\trial 3\Project3.exe

2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term
4. Find the key terms which are frequently searched. Display the number of times each key term
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : algorithm

partial matched keys

Algorithms

AlgoC

choose key Algorithms

Book Name (Key Term) : Algorithms

Book description : This is a book on advanced algorithms by cormen 2009 edition

number of pages : 1024

frequency it was searched for : 3

1. Insert into the Book index with appropriate key term, description and page number if a new term is found.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

frequency it was searched for : 5

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key terms displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the keys which have a partial match.
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : algorithm

partial matched keys

Algorithms

AlgoC

choose key



Type here to search



```
D:\dev\24 april\trial 3\Project3.exe
```

```
choose key DsJava
```

```
Book Name (Key Term) : DsJava
```

```
Book description : this is a book on implementation of datastructures in java
```

```
number of pages : 167
```

```
frequency it was searched for : 5
```

1. Insert into the Book index with appropriate key term, description and page number if a new term comes
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should
4. Find the key terms which are frequently searched. Display the number of times each key term has been
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropri
7. Exit

```
Enter option 6
```

```
enter a string, partial match occurs if 3 letters match : -
```



Type here to search



D:\dev\24 april\trial 3\Project3.exe

3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate option, the program should display all the keys matching the input.
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : DsJupyter

partial matched keys

DsJava

choose key DsJava

Book Name (Key Term) : DsJava

Book description : this is a book on implementation of datastructures in java

number of pages : 167

frequency it was searched for : 5

1. Insert into the Book index with appropriate key term, description and page number if a new term comes up.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

choose key AlgoC

Book Name (Key Term) : AlgoC
Book description : This is a book on basic algorithms in c
number of pages : 109
frequency it was searched for : 3

1. Insert into the Book index with appropriate key term, description and page number if a new term comes.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate key term, the system should display all the keys which have a partial match.
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : DsJupyter

partial matched keys

DsJava

choose key



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the term, the program has to display the details of the book.
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : algos

partial matched keys

Algorithms

AlgoC

choose key AlgoC

Book Name (Key Term) : AlgoC

Book description : This is a book on basic algorithms in c

number of pages : 109

frequency it was searched for : 3



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term comes.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should splay to the front.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched in descending order.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate term from the list.
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : algos

partial matched keys

Algorithms

AlgoC

choose key



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term has been used.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate term, the system should display the details.
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match :



Type here to search



D:\dev\24 april\trial 3\Project3.exe

2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and
4. Find the key terms which are frequently searched. Display the number of times each
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user
7. Exit

Enter option 6

enter a string, partial match occurs if 3 letters match : javascript

partial matched keys

DsJava

choose key DsJava

Book Name (Key Term) : DsJava

Book description : this is a datastructures book in java

number of pages : 167

frequency it was searched for : 4

1. Insert into the Book index with appropriate key term, description and page number
2. Delete the entry from the book index if the term is no longer needed.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if required.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the page number displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term is found.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses this option, the user has to enter the partial string.
7. Exit

Enter option 5

inorder traversal :

AlgoC Algorithms DsJava

preorder traversal :

DsJava Algorithms AlgoC

postorder traversal :

AlgoC Algorithms DsJava

1. Insert into the Book index with appropriate key term, description and page number if required.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and displayed.
4. Find the key terms which are frequently searched. Display the number of times each term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user enters a key term, all the key terms which have a partial match for the entered term should be displayed.
7. Exit

Enter option 3

enter key(book name) to seach for : DsJava

Book Name (Key Term) : DsJava

Book description : this is a datastructures book in java

number of pages : 167

frequency it was searched for : 3

1. Insert into the Book index with appropriate key term, description and page number.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and displayed.
4. Find the key terms which are frequently searched. Display the number of times each term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user enters a key term, all the key terms which have a partial match for the entered term should be displayed.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if required.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the page number displayed.
4. Find the key terms which are frequently searched. Display the number of times each term is found.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user enters the first three letters of the key term.
7. Exit

Enter option 5

inorder traversal :

AlgoC Algorithms DsJava

preorder traversal :

Algorithms AlgoC DsJava

postorder traversal :

AlgoC DsJava Algorithms

1. Insert into the Book index with appropriate key term, description and page number if required.
2. Delete the entry from the book index if the term is no longer needed.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term is found.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate term, the system should display the details.
7. Exit

Enter option 4

sorted by frequencies:

DsJava has been searched for 2 times

AlgoC has been searched for 2 times

Algorithms has been searched for 1 times

1. Insert into the Book index with appropriate key term, description and page number if a new term is found.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched.
5. Suitable display operations for every operation stated above.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term.
4. Find the key terms which are frequently searched. Display the number of times each key term.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the term.
7. Exit

Enter option 3

enter key(book name) to seach for : Algorithms

Book Name (Key Term) : Algorithms

Book description : this is a book on advanced algorithms by cormen 2009 edition

number of pages : 1024

frequency it was searched for : 1

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term.
4. Find the key terms which are frequently searched. Display the number of times each key term.
5. Suitable display operations for every operation stated above.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, all the key terms which have a partial match for 3 letters will be displayed.
7. Exit

Enter option 3

enter key(book name) to seach for : DsJava

Book Name (Key Term) : DsJava

Book description : this is a datastructures book in java

number of pages : 167

frequency it was searched for : 2

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term comes.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should splay to the front.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched in descending order.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate term from the list.
7. Exit

Enter option 4

sorted by frequencies:

```
AlgoC      has been searched for    2      times
DsJava     has been searched for    1      times
Algorithms  has been searched for   0      times
```

1. Insert into the Book index with appropriate key term, description and page number if a new term comes.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should splay to the front.
4. Find the key terms which are frequently searched. Display the number of times each key term has been searched in descending order.



Type here to search



D:\dev\24 april\trial 3"\Project3.exe

1. Insert into the Book index with appropriate key term, description and page no.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved.
4. Find the key terms which are frequently searched. Display the number of times.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user enters the key term.
7. Exit

Enter option 3

enter key(book name) to search for : AlgoC

Book Name (Key Term) : AlgoC

Book description : this is a book on basic algorithms in C

number of pages : 109

frequency it was searched for : 2

1. Insert into the Book index with appropriate key term, description and page no.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved.
4. Find the key terms which are frequently searched. Display the number of times.
5. Suitable display operations for every operation stated above.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the key terms which have a partial match for the chosen key term.
7. Exit

Enter option 3

enter key(book name) to search for : AlgoC

Book Name (Key Term) : AlgoC

Book description : this is a book on basic algorithms in C

number of pages : 109

frequency it was searched for : 1

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the key terms which have a partial match for the chosen key term.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term.
4. Find the key terms which are frequently searched. Display the number of times each key term.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the term.
7. Exit

Enter option 3

enter key(book name) to seach for : DsJava

Book Name (Key Term) : DsJava

Book description : this is a datastructures book in java

number of pages : 167

frequency it was searched for : 1

1. Insert into the Book index with appropriate key term, description and page number if a new term.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term.
4. Find the key terms which are frequently searched. Display the number of times each key term.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the term.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term has been used.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate option, the program should display all the key terms which have a partial match for the given 3 letters.
7. Exit

Enter option 5

inorder traversal :

AlgoC Algorithms DsJava

preorder traversal :

Algorithms AlgoC DsJava

postorder traversal :

AlgoC DsJava Algorithms

1. Insert into the Book index with appropriate key term, description and page number if a new term.
2. Delete the entry from the book index if the term is no longer needed.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term is found.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the key terms which have a partial match for the chosen key term.
7. Exit

Enter option 1

Enter book details

Book Name (Key Term) : Algorithms

Book description : this is a book on advanced algorithms by cormen 2009 edition

number of pages : 1024

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term is found.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the key terms which have a partial match for the chosen key term.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key terms displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term is used.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the option, he has to enter the partial string.
7. Exit

Enter option 5

inorder traversal :

AlgoC DsJava

preorder traversal :

DsJava AlgoC

postorder traversal :

AlgoC DsJava

1. Insert into the Book index with appropriate key term, description and page number if a new entry.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, it has to be displayed.
7. Exit

Enter option 2

enter key(book name) to delete : JsDp

Book Name (Key Term) : JsDp

Book description : this is a dynamic programming tutorial book in javascript

number of pages : 167

frequency it was searched for : 1

deletion of this node was sucessful

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term has to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term was searched.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new term is found.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key term should be highlighted.
4. Find the key terms which are frequently searched. Display the number of times each key term has been used.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses the appropriate term, the system should display the details.
7. Exit

Enter option 5

inorder traversal :

AlgoC DsJava JsDp

preorder traversal :

JsDp DsJava AlgoC

postorder traversal :

AlgoC DsJava JsDp

1. Insert into the Book index with appropriate key term, description and page number if a new term is found.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key terms displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term is used.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the key terms which have a partial match for the chosen term.
7. Exit

Enter option 1

Enter book details

Book Name (Key Term) : JsDp

Book description : this is a dynamic programming tutorial book in javascript

number of pages : 167

1. Insert into the Book index with appropriate key term, description and page number if a new entry.
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the key terms displayed.
4. Find the key terms which are frequently searched. Display the number of times each key term is used.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user chooses a key term, display all the key terms which have a partial match for the chosen term.



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number if a
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the ke
4. Find the key terms which are frequently searched. Display the number of times each key
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user choose
7. Exit

Enter option 1

Enter book details

Book Name (Key Term) : AlgoC

Book description : this is a book on basic algorithms in C

number of pages : 109

1. Insert into the Book index with appropriate key term, description and page number if a
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and the ke
4. Find the key terms which are frequently searched. Display the number of times each key
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user choose



Type here to search



D:\dev\24 april\trial 3\Project3.exe

1. Insert into the Book index with appropriate key term, description and page number
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each term is found.
5. Suitable display operations for every operation stated above.
6. Display of all key terms if a partial match occurs for 3 letters - Once the user enters the first three letters of the key term.
7. Exit

Enter option 1

Enter book details

Book Name (Key Term) : DsJava

Book description : this is a datastructures book in java

number of pages : 167

1. Insert into the Book index with appropriate key term, description and page number
2. Delete the entry from the book index if the term is no longer needed.
3. If key term is given as input, corresponding description has to be retrieved and to be displayed.
4. Find the key terms which are frequently searched. Display the number of times each term is found.



Type here to search



