Instantly share code, notes, and snippets.

douglas-vaz / **graph_search.cpp**

Created 6 years ago

Breadth First Search and Depth First Search in C++

graph_search.cpp

```cpp
#include <iostream>
#include <string>
#include <vector>
#include <queue>
#include <stack>
#include <algorithm>

using namespace std;

class Node{
	char value;
	vector<Node> children;
public:
	Node(char c){
		value = c;
	}

	void addChild(Node n){
		children.push_back(n);
		return;
	}

	void addChild(char n){
		Node foo(n);
		children.push_back(foo);
	}

	char getValue(){
		return value;
	}

	vector<Node> getChildren(){
		return children;
	}

	bool isLeaf(){
		return children.size()==0;
	}

	bool operator==(Node b){
		return b.value==value;
	}
};


void construct(Node *r)
{
	string foo;
	cout<<"Enter children for "<< r->getValue() <<" (-1 for leaf)"<<endl;
	cin>>foo;

	if(foo == "-1")
		return;
	else{
```

```
		for(int i = 0; i < foo.length(); i++)
		{
			Node t(foo[i]);
			construct(&t);

			r->addChild(t);
		}
	}
}

string breadthFirstSearch(Node root, Node goal)
{
	std::queue<Node> Q;
	std::vector<Node> children;
	string path = "";

	Q.push(root);

	while(!Q.empty())
	{
		Node t = Q.front();
		path += t.getValue();

		Q.pop();

		if(t == goal){
			return path;
		}
		children = t.getChildren();
		for (int i = 0; i < children.size(); ++i)
		{
			Q.push(children[i]);
		}
	}
	return path;
}

string depthFirstSearch(Node root, Node goal)
{
	std::stack<Node> Q;
	std::vector<Node> children;
	string path = "";

	Q.push(root);

	while(!Q.empty())
	{
		Node t = Q.top();
		path += t.getValue();

		Q.pop();

		if(t == goal){
			return path;
		}
		children = t.getChildren();
		std::reverse(children.begin(),children.end());

		for (int i = 0; i < children.size(); ++i){
			Q.push(children[i]);
		}
	}
	return path;
}

int main(int argc, char** args)
{
```

```
122         char r;
123         cout<<"Enter root node"<<endl;
124         cin>>r;
125 
126         Node root(r);
127         construct(&root);
128 
129         cout<<"Enter Node to search for: ";
130         cin>>r;
131 
132         cout<<endl;
133 
134         cout<<"BFS Traversal: "<<breadthFirstSearch(root, Node(' '))<<endl;
135         cout<<"BFS Search Path: "<<breadthFirstSearch(root, Node(r))<<endl<<endl;
136 
137         cout<<"DFS Traversal: "<<depthFirstSearch(root, Node(' '))<<endl;
138         cout<<"DFS Search Path: "<<depthFirstSearch(root, Node(r))<<endl;
139 
140         return 0;
141 }
```

bdawco commented on Apr 22, 2016

Hi I need help!!
can any body explain me the DSF program?

asam139 commented on Feb 5

Perfect!!! Thanks so much +1: