



Hacettepe University

BBM 203

Software Laboratory I

## **ASSIGNMENT 4**

Fatih OKSUZOGLU

21328284

# 1-Problem Definition

In computer science, BackusNaur Form or BNF is one of the main representation techniques commonly used to define the syntax of languages such as computer programming languages, document formats, command sets, communication protocols, and the like. We are asked to write the C type commands in this assignment. We need to use Linked list structure and recursive structure. the important thing is that the tree is dynamic and the response time of the code is low. Random operations are also expected to follow mathematical rules in output.

## 2-Methods and solution

**Instalization of Tree :** I send the node name that will occur when creating the tree to the function that will occur each time. First I select the type of node and then add it to the tree. When I find the node with a single child, I specify one of the operations that I read under the node. The name data I hold in the nodes indicates the name of the nodes with children. The name variable in the childless node holds the operation name.

**Readingfile :** I read the data in the input file with realloc .I have defined them all with a function that returns this data as char array (string).

**Random integers :** In recursive functions, a random number-calling function remains attached to a single number. Calling the srand () function to the main function will solve the problem.

**Run:** Before executing the code, I created a node named root containing the "alg." terminal item and built the tree on top of it. Then I sent the root to the function that prints the leaves in turn, which I call Print Leafs.

**Printing Leafs of Tree:** The print function is printed by recursive nodes. If the node is empty, it returns, and if it is full, it walks its children. In this way, I convert the incoming void pointer variable to a type that is common to all nodes. Then I send the children back to the recursive node.

## 3-Functions implementation

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <time.h>
5  struct Node{
6      char* name;
7      struct Node*right ,*down;
8  };
9  struct Node* newNode(char* name){
10     struct Node* temp=(struct Node*)malloc(sizeof(struct Node));
11     temp->name = name;
12     temp->right = temp->down = NULL;
13     return temp;
14 }
15 int randint(int n) {
16     return rand()%n;
17 }
18 char * returnerindexinchar(char* ch){
19     return ch;
20 }
21 int printer(char*op_out,char*pre_out,char*rel_out,char*set_out,char*var_out,char*name){
22     printf("%s\n",name);
23     return 0;
24 }
25 void printLeafNodes(struct Node *temp,char*op_out,char*pre_out,char*rel_out,
26                     char*set_out,char*var_out){
27     if(temp->right==NULL && temp->down==NULL){
28         printer(op_out,pre_out,rel_out,set_out,var_out,temp->name);
29     }
30     if(temp->right!=NULL){
31         printLeafNodes(temp->right,op_out,pre_out,rel_out,set_out,var_out);
32     }
33     if(temp->down!=NULL){
34         printLeafNodes(temp->down,op_out,pre_out,rel_out,set_out,var_out);
35     }
36 }
37 struct Node* Createleaf(struct Node* root,int randomvalue,int itter){
38     struct Node* temp=newNode(root->name);
39     temp->right=root->right;
40     temp->down=root->down;
41     return temp;
42 }
43 int Run(char*op_out,char*pre_out,char*rel_out,char*set_out,char*var_out){
44     return 0;
45 }
46 char *readingfile(FILE *f){
47     char*ch;
48     while((ch=fgetc(f))!=EOF){
49         printf("%c",ch);
50     }
51     return ch;
52 }
53 int main(int argc, char *argv[]){
54     return 0;
55 }
```

## REFERENCES

[1] "Binary Tree Implementation".

<https://www.geeksforgeeks.org/binary-tree-set-1-introduction/> [Accessed: Jan 7, 2019].

[2] "2-3 Tree Implementation".

<https://www.geeksforgeeks.org/2-3-trees-search-and-insert/> [Accessed: Jan 7, 2019].

[3] "Recursive implementation in Linked List".

<https://www.geeksforgeeks.org/recursive-insertion-and-traversal-linked-list/> [Accessed: Jan 7, 2019].

[4] "Modify contents linked list".

<https://www.geeksforgeeks.org/modify-content-linked-list/> [Accessed: Jan 8, 2019].

[5] "Dinamik Bellek Yönetimi".

<http://www.cemdemir.net/c-programlama-dili/dinamik-bellek-yonetimi-321.html>

[Accessed: Jan 8, 2019].