

Laborator 4 - Arbori

Pentru acest laborator, cerințele problemelor includ citirea din linia de comandă a căilor fișierelor de intrare și de ieșire. Pentru aceasta, se poate folosi o implementare similară cu următoarea secvență de cod:

```
int main(int argc, char * argv[])
{
    //calea fisierului de intrare este data in argv[1]
    std::ifstream fin(argv[1]);
    //calea fisierului de iesire este data in argv[2]
    std::ofstream fout(argv[2]);
    //implementarea problemei
    //...

    fin.close();
    fout.close();
    return 0;
}
```

Problema 1 – codare Prüfer

Cerință

Implementați rutina de codare Prüfer. Implementarea trebuie să fie eficientă din punctul de vedere al memoriei folosite și al timpului de execuție.

Formatul sursei

Soluția va fi implementată în limbajul de programare C++. Numele sursei va fi „p1.cpp”. Căile fișierelor de intrare și de ieșire vor fi date ca parametri în linia de comandă, prima fiind calea fișierului de intrare și a doua fiind calea fișierului de ieșire.

Formatul fișierului de intrare

Fișierul de intrare conține pe prima linie un număr N . A doua linie conține N valori. Valoarea de pe poziția i reprezintă părintele nodului i . Pentru rădăcina arborelui, în loc de părinte se află valoarea -1 . Indexarea nodurilor se face de la 0.

Valorile din fișierul de intrare se încadrează în următoarele limite:

- $1 \leq N \leq 100000$.

Formatul fișierului de ieșire

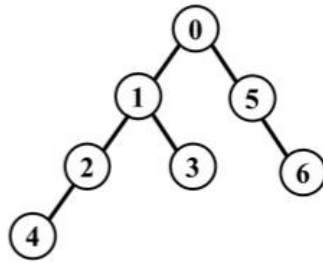
Fișierul de ieșire va conține două linii. Pe prima linie se va găsi numărul de valori din codarea Prüfer a arborelui, M . A doua linie va conține M valori separate prin spațiu, reprezentând codarea Prüfer.

Exemplu

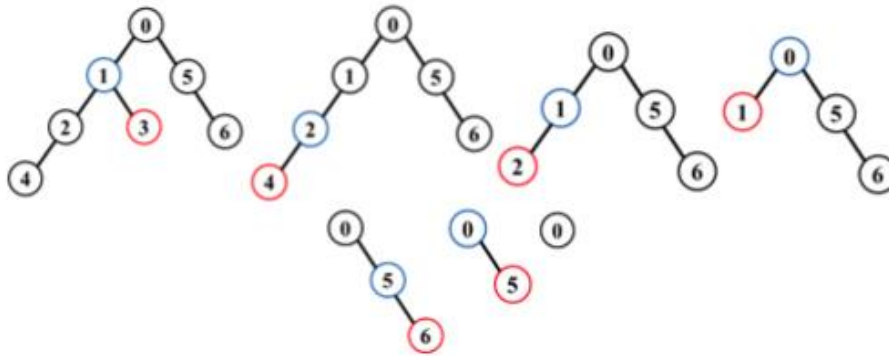
Fișier de intrare
7
-1 0 1 1 2 0 5

Fișier de ieșire
6
1 2 1 0 5 0

Fișierul de intrare descrie un arbore cu 7 noduri. Acest arbore poate fi vizualizat în figura următoare:



La fiecare pas al algoritmului de codare, se alege frunza cu indexul cel mai mic și se adaugă părintele ei în codare. Frunza este eliminată din arbore. Algoritmul continuă până când în arbore rămâne doar rădăcina. Pașii algoritmului de codare al arborelui dat pot fi vizualizați în figurile următoare. Cu roșu s-a marcat frunza aleasă, iar cu albastru nodul care este adăugat în codare.



Problema 2 – Decodare Prüfer

Cerință

Implementați rutina de decodare Prüfer. Implementarea trebuie să fie eficientă din punctul de vedere al memoriei folosite și al timpului de execuție.

Formatul sursei

Soluția va fi implementată în limbajul de programare C++. Numele sursei va fi „p2.cpp”. Căile fișierelor de intrare și de ieșire vor fi date ca parametri în linia de comandă, prima fiind calea fișierului de intrare și a doua fiind calea fișierului de ieșire.

Formatul fișierului de intrare

Fișierul de intrare conține două linii. Pe prima linie se găsește numărul de valori din codarea Prüfer a arborelui, M . A doua linie conține M valori separate prin spațiu, reprezentând codarea Prüfer. Indexarea nodurilor arborelui codat se face de la 0.

Valorile din fișierul de intrare se încadrează în următoarele limite:

- $1 \leq M \leq 100000$.

Formatul fișierului de ieșire

Fișierul de ieșire va conține pe prima linie un număr N , numărul de noduri ale arborelui. A doua linie va conține N valori. Valoarea de pe poziția i reprezintă părintele nodului i . Pentru rădăcina arborelui, în loc de părinte se afișează valoarea -1.

Exemplu

Fișier de intrare
6
1 2 1 0 5 0

Fișier de ieșire
7
-1 0 1 1 2 0 5

Problema 3 – Codare Huffman

Cerință

Să se codeze Huffman un text dat. La construcția arborelui folosit pentru stabilirea codurilor Huffman, prioritatea la alegerea nodurilor este frecvența, iar în caz de egalitate, se alege nodul în al cărui subgraf se găsește caracterul cu codul ASCII cel mai mic. Implementarea trebuie să fie eficientă din punctul de vedere al memoriei folosite și al timpului de execuție.

Formatul sursei

Soluția va fi implementată în limbajul de programare C++. Numele sursei va fi „p3.cpp”. Căile fișierelor de intrare și de ieșire vor fi date ca parametri în linia de comandă, prima fiind calea fișierului de intrare și a doua fiind calea fișierului de ieșire.

Formatul fișierului de intrare

Fișierul de intrare conține un text. Lungimea textului va fi de maxim 10000 de caractere.

Formatul fișierului de ieșire

Fișierul de ieșire va conține alfabetul folosit în codarea Huffman și textul codat.

Pe prima linie se va găsi un număr N reprezentând numărul de caractere din alfabet. Pe următoarele N linii se găsește descris alfabetul. Fiecare linie conține un caracter ch, un spațiu și un număr fr (fără alte spații albe). fr este frecvența caracterului ch în textul dat în fișierul de intrare. Alfabetul va fi afișat în ordinea lexicografică a caracterelor (în funcție de codul ASCII).

Pe următoarea linie se găsește un șir format din 0 și 1, reprezentând codarea Huffman a textului din fișierul de intrare.

Exemplu

Fișier de intrare
Loorrrreem

Fișier de ieșire
5
L 1
e 2
m 1
o 2
r 3
00010101111110101001

Problema 4 – Decodare Huffman

Cerință

Să se decodeze Huffman un cod dat. La construcția arborelui folosit pentru stabilirea codurilor Huffman, prioritatea la alegerea nodurilor este frecvența, iar în caz de egalitate, se alege nodul în al cărui subarbore se găsește caracterul cu codul ASCII cel mai mic. Implementarea trebuie să fie eficientă din punctul de vedere al memoriei folosite și al timpului de execuție.

Formatul sursei

Soluția va fi implementată în limbajul de programare C++. Numele sursei va fi „p4.cpp”. Căile fișierelor de intrare și de ieșire vor fi date ca parametrii în linia de comandă, prima fiind calea fișierului de intrare și a doua fiind calea fișierului de ieșire.

Formatul fișierului de intrare

Fișierul de intrare conține alfabetul folosit în codarea Huffman și textul codat.

Pe prima linie se găsește un număr N reprezentând numărul de caractere din alfabet. Pe următoarele N linii se găsește descris alfabetul. Fiecare linie conține un caracter ch , un spațiu și un număr fr (fără alte spații albe). fr este frecvența caracterului ch în textul codat. Alfabetul este scris în ordinea lexicografică a caracterelor (în funcție de codul ASCII). (Observație: printre caracterele din alfabet poate fi și caracterul spațiu.)

Pe următoarea linie se găsește un șir format din 0 și 1, reprezentând codarea Huffman a textului din fișierul de intrare.

Valorile din fișierul de intrare se încadrează în următoarele limite:

- $1 \leq N \leq 100$;
- $1 \leq fr \leq 1000$;
- $1 \leq L \leq 50000$ (lungimea șirului de biți).

Formatul fișierului de ieșire

Fișierul de ieșire conține textul decodat.

Exemplu

Fișier de intrare
5
L 1
e 2
m 1
o 2
r 3
00010101111110101001
Fișier de ieșire
Loorrem

Problema 5 – Arbore Minim de Acoperire

Cerință

Se dă un graf neorientat ponderat, găsiți un arbore minim de acoperire pentru graful dat. Graful dat este conex. Implementarea trebuie să fie eficientă din punctul de vedere al memoriei folosite și al timpului de execuție. (Implementați un algoritm de arbore minim de acoperire: Kruskal, Prim)

Formatul sursei

Soluția va fi implementată în limbajul de programare C++. Numele sursei va fi „p5.cpp”. Căile fișierelor de intrare și de ieșire vor fi date ca parametri în linia de comandă, prima fiind calea fișierului de intrare și a doua fiind calea fișierului de ieșire.

Formatul fișierului de intrare

Fișierul de intrare conține pe prima linie 2 numere separate prin spațiu: V E . V reprezintă numărul de noduri ale grafului. E reprezintă numărul de muchii ale grafului.

Următoarele E linii vor conține câte 3 numere separate prin spațiu, reprezentând câte o muchie: x y w . x și y sunt nodurile conectate de muchie, iar w este ponderea asociată muchiei. Indexarea vârfurilor se face de la 0.

Valorile din fișierul de intrare se încadrează în următoarele limite:

- $1 \leq V \leq 5000$;
- $0 \leq E \leq 50000$;
- $0 \leq x < V$;
- $0 \leq y < V$;
- $-1000 \leq w \leq 1000$.

Formatul fișierului de ieșire

Fișierul de ieșire conține pe prima linie un număr C reprezentând costul asociat arborelui minim de acoperire.

Pe a doua linie va conține un număr N reprezentând numărul de muchii al arborelui. Pe următoarele N linii vor fi descrise muchiile arborelui, prin cele două noduri pe care le conectează.

Exemplu

Fișier de intrare
5 8
0 2 6
0 3 7
0 4 9
1 3 4
1 4 5
2 3 9
2 4 1
3 4 6

Fișier de ieșire
16
4
0 2
1 3
1 4
2 4