МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС

«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»

Лабораторна робота № 4

з курсу «Теорія прийняття рішень»

Тема: «Прийняття рішень за допомогою методів голосування»

Виконав:

студент IV курсу

групи ДА-32

Колінько Анжела

Київ – 2016

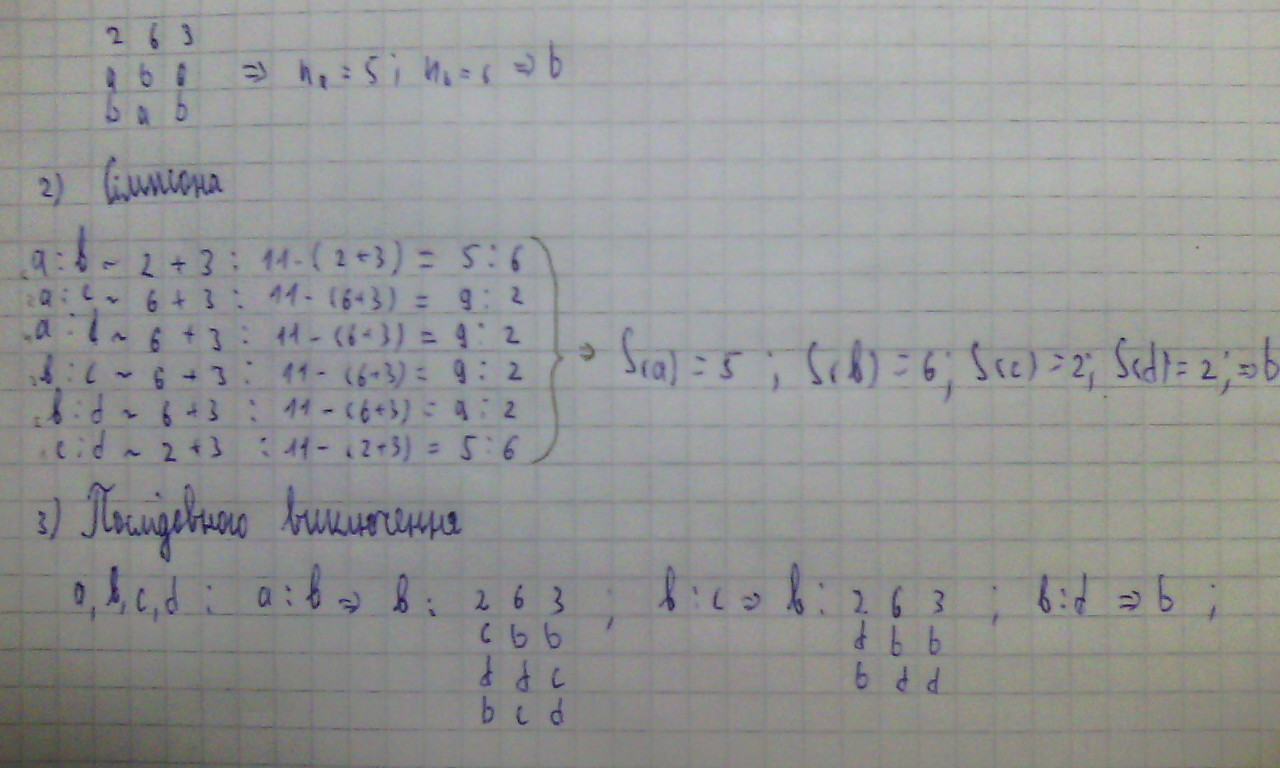
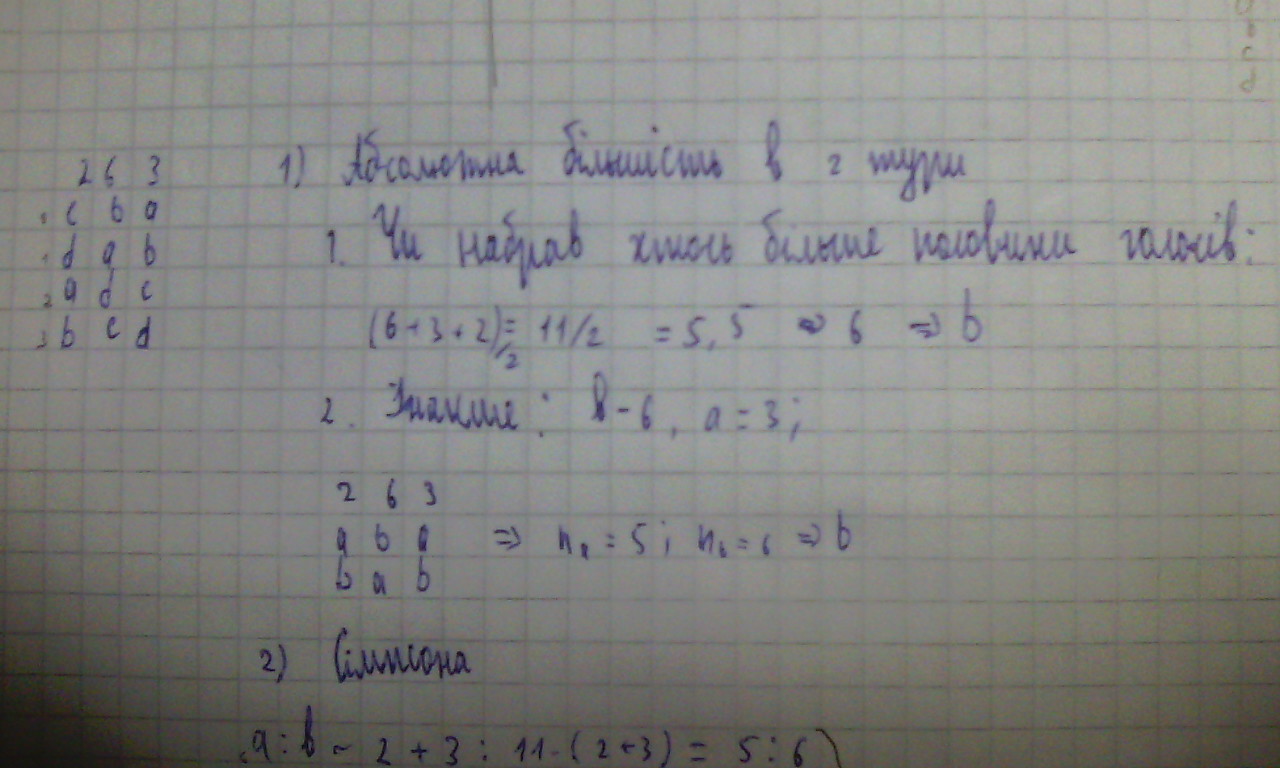
Варіант 11

Мета: ознайомитись з методами прийняття рішень за допомогою методів голосування.

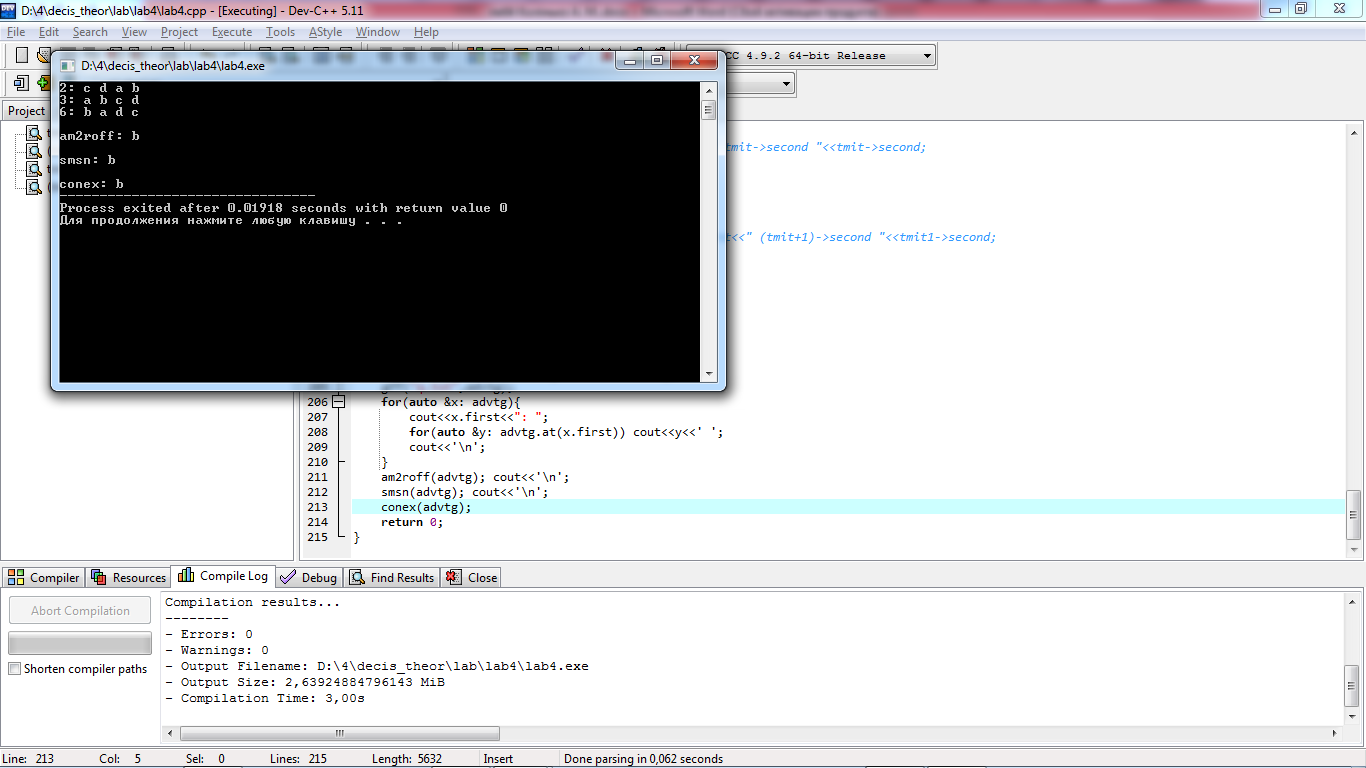
|  |  |
| --- | --- |
| Профіль | Методи |
| |  |  |  | | --- | --- | --- | | 2 | 6 | 3 | | c | b | a | | d | a | b | | a | d | c | | b | c | d | | - абсолютної більшості в два тури  - Сімпсона  - послідовного виключення |

1. Короткі теоретичні відомості:

Нехай  – множина "виборців", – множина "кандидатів". Кожен виборець задає "індивідуальну перевагу" на множині кандидатів у вигляді строгого ранжування, тобто задає лінійний порядок L(A).

2. Аналітичний розрахунок колективного ранжування:

3. Результати роботи програми:



Висновок: на лабораторній роботі було розглянуто методи прийняття рішень за допомогою методів голосування. Внаслідок значної переваги b, кожен з розглянутих методів в даному випадку дав однаковий результат.

Додаток. Лістинг

#include<iostream>

#include<string>

#include<fstream>

#include<vector>

#include<map>

#include<algorithm>

using namespace std;

#define nocdd 4

bool gff(string ifn, map<uint16\_t,vector<unsigned char>>& advtg){

ifstream fpi; uint8\_t cnt=0;

fpi.open(ifn.c\_str(),ios::binary);

if(!fpi.is\_open()) {cout<<"\nerr: gff no input file"; return 0;}

else{

uint16\_t novs; unsigned char ecd;

while(1){

if(cnt%5==0){

fpi>>novs;

cnt++;

}

else{

for(uint8\_t i=0;i<nocdd;i++){

fpi>>ecd;

advtg[novs].push\_back(ecd);

cnt++;

}

}

if(fpi.eof()) break;

}

fpi.close();

}

return 1;

}

rbm(map<uint16\_t,vector<unsigned char>>& namp, unsigned char x){

//cout<<"\nx="<<x;

for(auto &mp: namp){

//cout<<"\nbmp.second="; for(auto &vc: mp.second) cout<<vc<<' ';

mp.second.erase(find(mp.second.begin(),mp.second.end(),x));

//cout<<"\nabmp.second="; for(auto &vc: mp.second) cout<<vc<<' ';

}

}

void am2roff(map<uint16\_t,vector<unsigned char>>& advtg){

uint16\_t sum=0;

for(auto &x: advtg){

sum+=x.first;

}

for(auto &x: advtg){

if(x.first>=((sum+1)/2)){

cout<<"\nam2roff: "<<x.second.at(0)<<' ';//<<((sum+1)/2);

return;

}

}

unsigned char max1=advtg.begin()->second.at(0),max2=advtg.begin()->second.at(0);

uint16\_t tmpmax=0; map<unsigned char,uint16\_t> tmpmp;

for(uint8\_t i=0;i<nocdd;i++){

tmpmp[advtg.begin()->second.at(i)]=0;

for(auto &x: advtg){

if(x.second.at(0)==advtg.begin()->second.at(i))

tmpmp.at(advtg.begin()->second.at(i))+=x.first;

}

}

//for(auto& x: tmpmp) cout<<x.first<<": "<<x.second<<'\n';

for(auto& x: tmpmp){

if(tmpmax<x.second){

tmpmax=x.second;

max1=x.first;

}

}

tmpmax=0;

for(auto& x: tmpmp){

if(tmpmax<x.second&&max1!=x.first){

tmpmax=x.second;

max2=x.first;

}

}

map<uint16\_t,vector<unsigned char>> namp;

namp=advtg;

while(namp.begin()->second.size()!=2){

for(auto& x: namp.begin()->second){

if(max1!=x&&max2!=x){

rbm(namp,x);

break;

}

}

}

/\*for(auto &x: namp){

cout<<x.first<<": ";

for(auto &y: namp.at(x.first)) cout<<y<<' '; cout<<'\n';

}\*/

tmpmp.clear();

max1=namp.begin()->second.at(0); tmpmax=0;

for(auto &x: namp){

if(max1==x.second.at(0))

tmpmax+=x.first;

}

if(tmpmax>(sum-tmpmax)){

cout<<"\nam2roff: "<<max1;

return;

}

else{

cout<<"\nam2roff: "<<namp.begin()->second.at(1);

return;

}

}

void smsn(map<uint16\_t,vector<unsigned char>>& advtg){

map<vector<unsigned char>,vector<uint16\_t>> qts;

uint8\_t cntl=0,cntr=1;

uint16\_t sum=0;

for(auto &x: advtg){

sum+=x.first;

}

while(cntl!=3){

vector<unsigned char> uctmpv;

uint16\_t uitmp=0;

uctmpv.push\_back(advtg.begin()->second.at(cntl));

uctmpv.push\_back(advtg.begin()->second.at(cntr));

//for(auto &x: uctmpv) cout<<x<<' '; cout<<'\n';

for(auto &x: advtg){

if(find(x.second.begin(),x.second.end(),uctmpv.at(0))<

find(x.second.begin(),x.second.end(),uctmpv.at(1))){

uitmp+=x.first;

}

}

qts[uctmpv]={uitmp,(uint16\_t)(sum-uitmp)};

cntr++;

if(cntr==4){

cntl++; cntr=cntl+1;

}

}

/\*for(auto &x: qts){

for(auto &y: x.first){

cout<<y<<' ';

}cout<<": ";

for(auto &y: x.second){

cout<<y<<' ';

}

cout<<'\n';

}\*/

map<unsigned char,uint16\_t> smp;

for(auto &x: advtg.begin()->second){

smp[x]=sum;

}

//for(auto &x: smp) cout<<x.first<<": "<<x.second<<'\n';

for(auto &psmp: smp){

for(auto &x: qts){

if(find(x.first.begin(),x.first.end(),psmp.first)!=x.first.end()){

uint8\_t pos=find(x.first.begin(),x.first.end(),psmp.first)-x.first.begin();

if(x.second.at(pos)<psmp.second) psmp.second=x.second.at(pos);

}

}

}

//for(auto &x: smp) cout<<x.first<<": "<<x.second<<'\n';

uint16\_t mx=smp.begin()->second; unsigned char mxuc=smp.begin()->first;

for(auto &x: smp){

if(x.second>mx){

mx=x.second; mxuc=x.first;

}

}

cout<<"\nsmsn: "<<mxuc;

}

void conex(map<uint16\_t,vector<unsigned char>>& advtg){

map<uint16\_t,vector<unsigned char>> nadg=advtg;

map<unsigned char,uint16\_t> tmpmp;

for(uint8\_t i=0;i<nocdd;i++){

tmpmp[advtg.begin()->second.at(i)]=0;

for(auto &x: advtg){

if(x.second.at(0)==advtg.begin()->second.at(i))

tmpmp.at(advtg.begin()->second.at(i))+=x.first;

}

}

//for(auto& x: tmpmp) cout<<x.first<<": "<<x.second<<'\n';

map<unsigned char,uint16\_t>::iterator tmit=tmpmp.begin(),tmit1=tmpmp.begin(); ++tmit1;

while(nadg.begin()->second.size()!=1){

//cout<<"\n\n";

for(uint8\_t i=0;i<nadg.begin()->second.size();i++){

tmpmp.at(nadg.begin()->second.at(i))=0;

for(auto &x: nadg){

if(x.second.at(0)==nadg.begin()->second.at(i))

tmpmp.at(nadg.begin()->second.at(i))+=x.first;

}

}

//for(auto& x: tmpmp) cout<<x.first<<": "<<x.second<<'\n';

tmit=tmpmp.begin(),tmit1=tmpmp.begin(); ++tmit1;

if((tmit->second)<(tmit1->second)){

//cout<<"\ntmit->first "<<tmit->first<<" tmit->second "<<tmit->second;

rbm(nadg,tmit->first);

tmpmp.erase(tmit->first);

}

else{

//cout<<"\n(tmit+1)->first "<<tmit1->first<<" (tmit+1)->second "<<tmit1->second;

rbm(nadg, tmit1->first);

tmpmp.erase(tmit1->first);

}

}

cout<<"\nconex: "<<nadg.begin()->second.at(0);

}

int main(){

map<uint16\_t,vector<unsigned char>> advtg;

gff("q.txt",advtg);

for(auto &x: advtg){

cout<<x.first<<": ";

for(auto &y: advtg.at(x.first)) cout<<y<<' ';

cout<<'\n';

}

am2roff(advtg); cout<<'\n';

smsn(advtg); cout<<'\n';

conex(advtg);

return 0;

}