void main()

{

int CPU = 20;

for (int Alg = 0; Alg < 3; Alg++)

{

switch (Alg)

{

case 0:

{

vector<vector<int>>SpanFactory;

SpanFactory.resize(270);

for (int ins = 0; ins < SpanFactory.size(); ins++)

SpanFactory[ins].resize(5, 0);

for (int Ins = 0; Ins < 270; Ins++)

{

Read(Ins);

long TimeLimit = CPU\*gJobs\*gMachine;

for (int i = 0; i < 5; i++) //每个测试用例执行5遍

{

srand(i + 100 + Ins);

SpanFactory[Ins][i] = IGM(TimeLimit, 3, 10, 3); //不同的算法

cout << "Ins" << "\t" << Ins << "\t" << "rep" << "\t" << i << "\t" << "IGM" << endl;

}

}

string FileDirectory = "Result\\"; //

ostringstream str;

str << "IGM\_20" << ".txt"; //不同的算法

ofstream ofile;

ofile.open(FileDirectory + str.str());

for (int Ins = 0; Ins < SpanFactory.size(); Ins++)

{

for (int Rep = 0; Rep < SpanFactory[Ins].size(); Rep++)

{

ofile << SpanFactory[Ins][Rep] << "\t";

}

ofile << endl;

}

ofile.close();

}

case 1:

{

vector<vector<int>>SpanFactory;

SpanFactory.resize(270);

for (int ins = 0; ins < SpanFactory.size(); ins++)

SpanFactory[ins].resize(5, 0);

for (int Ins = 0; Ins <270; Ins++)

{

Read(Ins);

long TimeLimit = CPU\*gJobs\*gMachine;

for (int i = 0; i < 5; i++)

{

srand(i + 100 + Ins);

SpanFactory[Ins][i] = IG\_Compared(TimeLimit, 3, 0.2);

cout << "Ins" << "\t" << Ins << "\t" << "rep" << "\t" << i << "\t" << "IG\_Compared" << endl;

}

}

string FileDirectory = "Result\\"; //

ostringstream str;

str << "IG\_Compared\_20" << ".txt"; //不同的算法

ofstream ofile;

ofile.open(FileDirectory + str.str());

for (int Ins = 0; Ins < SpanFactory.size(); Ins++)

{

for (int Rep = 0; Rep < SpanFactory[Ins].size(); Rep++)

{

ofile << SpanFactory[Ins][Rep] << "\t";

}

ofile << endl;

}

ofile.close();

}

case 2:

{

vector<vector<int>>SpanFactory;

SpanFactory.resize(270);

for (int ins = 0; ins < SpanFactory.size(); ins++)

SpanFactory[ins].resize(5, 0); //每个算法每个算例执行5遍

for (int Ins = 0; Ins <270; Ins++)

{

Read(Ins);

long TimeLimit = CPU\*gJobs\*gMachine;

for (int i = 0; i < 5; i++) //每个测试用例执行5遍

{

srand(i + 100 + Ins);

SpanFactory[Ins][i] = CRO(TimeLimit, 150, 0.7, 10, 50); //不同的算法

cout << "Ins" << "\t" << Ins << "\t" << "rep" << "\t" << i << "\t" << "CRO" << endl;

}

}

string FileDirectory = "Result\\"; //

ostringstream str;

str << "CRO\_20" << ".txt"; //不同的算法

ofstream ofile;

ofile.open(FileDirectory + str.str());

for (int Ins = 0; Ins < SpanFactory.size(); Ins++)

{

for (int Rep = 0; Rep < SpanFactory[Ins].size(); Rep++)

{

ofile << SpanFactory[Ins][Rep] << "\t";

}

ofile << endl;

}

ofile.close();

}

}

}

}