1-1.図を描くことにより,例題２の解を２部グラフ(bipartite graph)上に示せ.



1-2.例題3を解いた解を(数式で記述するのではなく)文章で説明せよ.

　例題3における解のvalueの値が１か0の整数値に限定されておりこの値が1になっているvariableの組み合わせがこの問題の最適な解となっている.

1-3.例題4の問題内容を説明し,解いた解を(数式で記述するのではなく)文章で説明せよ.

　ある輸送会社は請求可能な料金まで限界に輸送用コンテナを積み,運びたい。広さの制約は1000立方フィートで重さは1200ポンドである.

　例題４における解のvalueの値が1か0の整数値に限定されおりこの値が１になっているvariableの組み合わせがこの問題の最適な解となっている.

1-4.例題2において,すべての輸送費が同じである場合,どのように供給ノード全体に品物を輸送すればよいのか?供給ノードiの供給量は正の整数,需要ノードjの需要量は正の整数とする.供給ノードの総数はm,需要ノードの総数はnとする.供給ノードiから需要ノードjへの輸送量をとする.解が存在するように,との値は

を満たすように設定すること.輸送量を求めるアルゴリズムを考え,そのプログラム・コードをC言語で書け.数値例を作成し,これを解き,その出力結果を添付せよ.m10,n8とする.

ソースファイル

#include<stdio.h>

#include<time.h>

#include<stdlib.h>

#define M 10

#define N 8

FILE \*fp;

int main(int argc, char\* argv[]){

int i=0, j=0, x;

int a[M] = { 30, 25, 21,16,50,31,12,15,56,21 };

int b[N] = { 15, 17, 22, 12,13,16,18,14 };

fopen\_s(&fp, "kekka.txt", "w");

while (i != M && j != N){

fprintf(fp, "a[%d]=%2d\td[%d]=%2d\t", i, a[i], j, b[j]);

if (a[i] < b[j]){

x = a[i];

b[j] -= a[i];

fprintf(fp, "x[%d][%d]=%2d\n", i, j, x);

i++;

}

else if (a[i]>b[j]){

x = b[j];

a[i] -= b[j];

fprintf(fp, "x[%d][%d]=%2d\n", i, j, x);

j++;

}

else if (a[i] == b[j]){

x = b[i];

fprintf(fp, "x[%d][%d]=%2d\n", i, j, x);

i++;

j++;

}

}

return 0;

}

出力結果

a[0]=30 d[0]=15 x[0][0]=15

a[0]=15 d[1]=17 x[0][1]=15

a[1]=25 d[1]= 2 x[1][1]= 2

a[1]=23 d[2]=22 x[1][2]=22

a[1]= 1 d[3]=12 x[1][3]= 1

a[2]=21 d[3]=11 x[2][3]=11

a[2]=10 d[4]=13 x[2][4]=10

a[3]=16 d[4]= 3 x[3][4]= 3

a[3]=13 d[5]=16 x[3][5]=13

a[4]=50 d[5]= 3 x[4][5]= 3

a[4]=47 d[6]=18 x[4][6]=18

a[4]=29 d[7]=14 x[4][7]=14

min 3x12+4x13+2x23+5x25+3x34+6x36+7x47+3x56+6x58+5x64+3x69+10x79+9x710+5x86+4x811+6x911+8x1110

st

node1) x12+x13=10

node2) x23+x25-x12=-1

node3) x34+x36-x13-x23=-1

node4) x47-x34-x64=-1

node5) x56+x58-x25=-1

node6) x64+x69-x36-x56-x86=-1

node7) x79+x710-x47=-1

node8) x86+x811-x58=-1

node9) x911-x69-x79=-1

node10) -x710-x1110=-1

!node11) x1110-x911-x811=-1

end







max v

st

node1) x12+x14-v=0

node2) x23+x24-x12=0

node3) x35+x36-x23=0

node4) x47+x48-x14-x24=0

node5) x75-x47=0

node6) x89-x48=0

node7) x59-x75-x35=0

node8) x69-x36=0

arc12) x12<=4

arc14) x14<=10

arc24) x24<=4

arc23) x23<=2

arc47) x47<=5

arc69) x69<=4

arc75) x75<=4

arc89) x89<=11

arc38) x38<=8

arc36) x36<=5

arc48) x48<=5

end

