חיתוך הבד/תרגיל מדהים ויעילות אלופה!!!!!!!!!

namespace material

{

class Program

{

public static int BestPrice(int[,] mat,int height,int width)

{

int a = 0, b = 0;

int[,] help = new int[height + 1, width + 1];

help[1, 1] = mat[1, 1];

for(int i=2;i<9;i++)

{

for(int j=1;j<i&j<5;j++)

{

Console.WriteLine((j) + " " + (i-j));

if (j % 2 == 0)

{

if ((i-j) % 2 == 0)//2\*2

{

a = j / 2;

b = (i - j) / 2;

if (help[j,b] \* 2 > help[a,i-j] \* 2)

help[j, i - j] = help[j, b] \* 2;

else

help[j, i - j] = help[a, i - j] \* 2;

//axi-j\*2||jxb\*2

}

else//2\*5

{

a = j / 2;

//axi-j \*2||

b = (i - j - 1) / 2;

//jxb\*2+j\*1

if ((help[a, i-j] \* 2) < ((help[j,b] \* 2)+(help[j,1])))

help[j, i - j] = ((help[j, b] \* 2) + (help[j, 1]));

else

help[j, i - j] = (help[a, i - j] \* 2);

}

}

else

{

if ((j + i) % 2 == 0)//5\*4

{

a = (i - j) / 2;

b = (j - 1) / 2;

//jxa\*2|bxi-j \*2 +1xi-j

if ((help[j,a] \* 2) < ((help[b,i-j] \* 2) + (help[1,i-j])))

help[j, i - j] = ((help[b, i - j] \* 2) + (help[1, i - j]));

else

help[j, i - j] = (help[j, a] \* 2);

}

else//5\*5 by this all of them

{

a = (j - 1) / 2;

b = (i - j - 1) / 2;

//axi-j \* 2+ 1\*i-j||

//jxb \*2+j\*1

if (((help[j, b\* 2] ) + (help[j, 1])) < ((help[a\* 2, (i - j) ]) + (help[1, i - j])))

help[j, i - j] = (((help[a\* 2, (i - j)]) + (help[1, (i - j) ])));

else

help[j, i - j] = ((help[ j,b\* 2] ) + (help[ j,1]));

}

}

if (help[j, i - j] < mat[j, i - j])

help[j, i - j] = mat[j, i - j];

Console.WriteLine(help[j, i - j] + " ");

}

}

return mat[4,7];

}

static void Main(string[] args)

{

int[,] mat = {{0,0,0,0,0,0,0,0,0,0},{0,1,4,5,0,3,0,4,2,1},{0,8,2,1,0,0,0,5,1,3}

,{0,0,0,0,2,5,7,2,4,2 },{0,10,12,0,5,1,0,0,0,9 },{0,3,4,7,2,0,5,2,0,0 },{0,0,3,7,4,2,1,1,1,8 },{0,2,3,1,2,3,0,3,1,2 }};

Console.WriteLine("the most practious is "+BestPrice(mat,4,7));

}

}

}

תרגיל יפהיפה למציאת תת מערך השווה לסכום מסוים במדויק והדפסת הערכים שהובילו לתוצאה:

public static bool SubExactually(int[] arr, int num)

{

bool[,] mat = new bool[arr.Length + 1, num + 1];//all the array is false

for (int i = 0; i < arr.Length + 1; i++)

mat[i, 0] = true;

for (int i = 1; i < arr.Length + 1; i++)

{

for (int j = 1; j < num + 1; j++)

{

if (mat[i - 1, j] == true)

mat[i, j] = true;

else

{

if ((arr[i - 1] > j) || (mat[i - 1, (j - arr[i - 1])] == false))

mat[i, j] = false;

else

mat[i, j] = true;

}

}

}

for (int i = 0; i < arr.Length + 1; i++)

{

for (int j = 0; j < num + 1; j++)

{

if (mat[i, j] == true)

Console.Write("true ");

else

Console.Write("fals ");

}

Console.WriteLine(" ");

}

int ii = arr.Length, jj = num;

while (ii > 1 && jj > 1)

{

if (mat[ii, jj] == true)

{

if (mat[ii - 1, jj] == false)

{

Console.WriteLine(arr[ii - 1]);

jj = jj - arr[ii - 1];

arr[ii - 1] = 0;

}

else

ii--;

}

else

jj--;

}

if ((ii == 1) && (mat[ii, jj] == true))

{

Console.WriteLine(arr[0]);

arr[ii - 1] = 0;

}

return mat[arr.Length, num];

}

הפלט:

true fals fals fals fals fals fals fals fals

true fals true fals fals fals fals fals fals

true fals true fals true fals true fals fals

true fals true fals true fals true fals fals

true fals true fals true fals true fals true

6

2

True

For:: int[] w = { 2, 4, 33, 6 }; Console.WriteLine(SubExactually(w, 8));

ווריאציה נוספת של התרגיל, כאשר מעונינים לחלק את המערך לשלושה חלקים שווים במדויק:

public static bool ThreeSubSame(int[] arr)

{

int sum = 0;

for (int i = 0; i < arr.Length; i++)

sum += arr[i];

if (sum % 3 != 0)

return false;

else

{

if (SubExactually(arr, sum / 3) == true)

if (SubExactually(arr, sum / 3) == true)

return true;

}

return false;

}