Розв»язання задачі І

1)Знаходження максимальної суми

Для знаходження максимальної суми створимо 2-вимірний масив dp\_val такий, що значення dp\_val[i][j] рівне максимальній сумі, яку можна набрати, потрапивши у клітинку (i,j). Для клітинок першого рядка це значення рівне числу, що записане у самій клітинці(клітинки з першого рядка завжди початкові). Отже, dp\_val[1][i]=m[1][i], де m[i][j]-число, записане у клітинці (і,j).

У інші клітинки ми можемо потрапити з однієї з «верхніх сусідніх» клітинок, отже максимальна сума, яку можна набрати, потрапивши у (і,j) рівна сумі m[i][j] і максимуму з сум, що можна набрати, потрапивши у «верхні сусідні» клітинки.

Отже, dp[i][j]=m[i][j]+ Max( dp\_val[i-1][j-1], dp\_val[i-1][j] ,dp\_val[i-1][j+1] ) (проте, якщо j=1 або j=N(N-ширина таблиці), то у клітинки лише 2 «верхні сусіди» і це враховується у формулі)

Коли ми знайдемо значення dp\_val для всіх клітинок таблиці, залишається знайти максимум зі значень dp\_val для клітинок останнього рядка(адже, за умовою, шлях закінчується у одній з клітинок останнього рядка). Цей максимум і буде шуканою максимальною сумою.

2)Знаходження кількості шляхів

Для знаходження кількості шляхів створимо 2-вимірний масив dp\_way такий, що значення dp\_way[i][j] рівне кількості способів потрапити у клітинку (i,j), набравши максимальну можливу суму. Для клітинок першого рядка це значення рівне 1(бо у будь-яку клітинку з першого рядка можна потрапити рівно 1 способом – почати з неї шлях).

Для інших клітинок спершу знайдемо максимум із сум, що можна набрати, потрапивши у «верхні сусідні» клітинки. Нехай цей максимум рівний певному числу etalon. Тоді, щоб набрати максимальну можливу суму, у поточну клітинку потрібно прийти з однієї з тих «верхніх сусідніх» клітинок, максимальна сума для якої рівна etalon. А кількість способів потрапити у дану поточну клітинку рівна сумі кількостей способів потрапити у ті «верхні сусідні» клітинки, максимальна сума для яких рівна etalon. Тому, знаходження значення dp\_way[i][j] виглядає приблизно так:

etalon=Max(dp\_val[i-1][j-1],Max(dp\_val[i-1][j],dp\_val[i-1][j+1]));

dp\_val[i][j]=m[i][j]+etalon;

dp\_way[i][j]=0;

if(dp\_val[i-1][j-1]==etalon) dp\_way[i][j]+=dp\_way[i-1][j-1];

if(dp\_val[i-1][j]==etalon) dp\_way[i][j]+=dp\_way[i-1][j];

if(dp\_val[i-1][j+1]==etalon) dp\_way[i][j]+=dp\_way[i-1][j+1];

Коли ми знайдемо значення dp\_way для всіх клітинок таблиці, залишається знайти суму значень dp\_way для тих клітинок останнього рядка, для ких значення dp\_val рівне максимуму(адже, за умовою, шлях закінчується у одній з клітинок останнього рядка). Ця сума і буде шуканою кількістю шляхів.