Примерни решения на домашно № 2

Задача 1: Намерете асимптотичната сложност на всеки от следните три фрагмента от програми като функция на n.

```
int f(int n) {
 int i, s=2, m=n*n;
 for(i=0; i<m*n; i++)
  s += s;
 return s; }
```

```
int g(int n) {
 if(n < 10) return 1;
 int j=6, s=0;
 while(j > 8) {
  s += g(n-2);
  j --;}
 while(n-j > 1) {
  j = n;
  s += g(n-1) + g(n-2);}
 while(j >= n) {
  j = 2;
  s += g(n-j);}
 return s;}
```

```
int h(int n) {
 int i, t=0;
 if(n < 2) return 2;
 t += h(n/3);
 for(i = 2; i < n; i *= 2)
  t ++;
 t *= h(n/3);
 return t; }
```

Решения: Сложността на $f(\,)$ е тривиално $\Theta(n^3)$.

Сложността на $g(\,)$ се намира чрез следните разсъждения: първият `while` не се изпълнява изобщо, вторият `while` се изпълнява точно веднъж, оттам имаме по едно викане `g(n-1)` и `g(n-2)`, третият `while` се изпълнява точно веднъж и имаме още едно викане на `g(n-2)`; следователно, рекурентното уравнение за сложността е $T(n) = T(n-1) + 2T(n-2) + 1$, което има решение $T(n) = \Theta(2^n)$.

Сложността на $h(\,)$ се намира чрез следните разсъждения: имаме точно две викания $h\left(\frac{n}{3}\right)$ и освен това логаритмична работа (цикълът `for`); следователно, рекурентното уравнение за сложността е $T(n) = 2T\left(\frac{n}{3}\right) + \lg n$, което има решение $T(n) = \Theta\left(n^{\lg_3 2}\right)$.

Задача 2: Нека $a_1\ a_2\ \cdots\ a_n$ е пермутация на множеството $\{1, 2, \ldots, n\}$. *Инверсия* в тази пермутация се нарича всяка наредена двойка (i, j), такава че $1 \leq i < j \leq n$ и $a_i > a_j$. *Инверсният вектор на пермутацията* $a_1\ a_2\ \cdots\ a_n$ е векторът $(b_1, b_2, \ldots, b_n)$, където $\forall i, 1 \leq i \leq n$, b_i е броят на елементите в $a_1\ a_2 \cdots\ a_n$, които са вляво от i и са по-големи от i.

5 т. **а)** Напишете инверсния вектор на пермутацията 4 2 3 7 1 8 5 9 6.

25 т. **б)** Предложете колкото може по-ефикасен алгоритъм, който по зададен инверсен вектор извежда оригиналната пермутация. Допуснете, че входът $(b_1, b_2, \ldots, b_n)$ на алгоритъма е коректен инверсен вектор на някоя пермутация на числата $\{1, 2, \ldots, n\}$. Дайте кратка обосновка на коректността на Вашия алгоритъм (не се иска строго доказателство по индукция или с инвариант) и изследвайте сложността му по време.

Решение, а) $(4, 1, 1, 0, 2, 3, 0, 0, 0)$.

Решение, б) Да разсъждаваме така. Елемент k от пермутацията, която искаме да построим, може да има между 0 и $n - k$ числа вляво от себе си и по-големи от себе си в тази пермутация.

- Знаем, че числото n е елемент на пермутацията (която искаме да построим). За b_n има само една възможност – да бъде нула. Записваме n в списък.
- За b_{n-1} има две възможности. Ако b_{n-1} е нула, слагаме $n - 1$ вляво от n в списъка. В противен случай, вдясно от n в списъка.
- За b_{n-2} има три възможности. Ако b_{n-2} е нула, слагаме $n - 2$ вляво от вече сложените два елемента в списъка. Тоест, в началото. Ако b_{n-2} е единица, слагаме $n - 2$ между първи и втория елемент. Ако е двойка, слагаме $n - 2$ вдясно от първи и втория.
- И така нататък.
- Накрая разглеждаме b_1. Слагаме 1 на такава позиция в списъка, че точно b_1 елемента са вляво от него.

И изобщо, k бива сложен в списъка след като числата n, $n - 1$, $\ldots$, $k + 1$ вече са в списъка, и то по такъв начин, че точно b_k вече сложени числа са вляво от него. Следният алгоритъм имплементира тази идея.

GeneratePermutation($(b_1, b_2, \ldots, b_n)$: inversion vector)
нека X е списък от числа
for $k \leftarrow n$ **downto** 1
    сложи k в X така, че b_k елемента да са вляво от него
return X

Коректността на алгоритъма следва от горните разсъждения. Сложността му по време е квадратична, тъй като външният **for**-цикъл се изпълнява точно n пъти, а вътрешният (имплицитен) цикъл се изпълнява в най-лошия случай k пъти, така че в най-лошия случай сложността е пропорционална на

$$\sum_{k=1}^{n}\sum_{j=1}^{k} 1 = \sum_{k=1}^{n} k = \Theta(n^2)$$

$\square$

Задача 3: Нека $M[1\,..\,n, 1\,..\,m]$ е масив от естествени числа. За целите на тази задача ще казваме, че M е *интересен*, ако е изпълнено

$$M[p,q] + M[s,t] \leq M[p,t] + M[s,q]$$

за $1 \leq p < s \leq n$ и $1 \leq q < t \leq m$. Ето пример за масив 7×5, който е интересен:

15	19	8	4	25
16	18	5	1	20
50	35	22	10	24
91	64	21	9	5
90	60	16	4	0
100	70	20	4	0
120	75	22	5	1

15 т. 1. Предложете алгоритъм с **линейна** сложност по време, чийто вход е масив $M[1\,..\,n, 1\,..\,m]$ и чийто изход е или 1, ако M е интересен, или 0, ако M не е интересен. Докажете коректността и сложността по време на този алгоритъм.

Забележка: сложността по време е функция от големината на входа.

5 т. 2. За $i \in \{1, 2, \ldots, n\}$, нека $\phi(i)$ е номерът на колоната, съдържаща най-левия минимален елемент в ред i на интересен масив $M[1\,..\,n, 1\,..\,m]$. Докажете, че

$$\phi(1) \leq \phi(2) \leq \cdots \leq \phi(n)$$

5 т. 3. Нека

$$n_e = \{x \in \{1, 2, \ldots, n\} \,|\, x \text{ е четно}\}$$
$$n_o = \{x \in \{1, 2, \ldots, n\} \,|\, x \text{ е нечетно}\}$$

Допуснете, че са дадени $\phi(i)$ за всички $i \in n_e$. Обяснете как да намерим $\phi(i)$ за всички $i \in n_o$ във време $O(n + m)$.

15 т. 4. Предложете алгоритъм със сложност по време $O(n + m \lg n)$, изграден по схемата **Разделяй-и-Владей**, който има вход интересен масив $M[1\,..\,n, 1\,..\,m]$ и който изчислява $\phi(i)$ за $i \in \{1, 2, \ldots, n\}$. Не е необходимо да пишете псевдокод, но трябва да обясните ясно и недвусмислено какво имате предвид. Накратко обосновете коректността и асимптотичната горна граница за сложността по време на Вашия алгоритъм.

Упътване: Вашето решение на 3) е добра отправна точка за конструиране на такъв алгоритъм.

Решение: Такъв масив се нарича *масив на Монж* (*Monge array*). Алгоритъм, който проверява дали даден масив е масив на Монж и е базиран директно на дефиницията “$M[p,q] + M[s,t] \leq M[p,t] + M[s,q]$ за $1 \leq p < s \leq n$ и $1 \leq q < t \leq m$”, е следният.

Test Mongeness 1($M[1\,..\,n, 1\,..\,m]$)

```
for p ← 1 to n − 1
    for s ← p + 1 to n − 1
        for q ← 1 to m − 1
            for t ← q + 1 to m
                if M[p, q] + M[s, t] > M[p, t] + M[s, q]
                    return 0
return 1
```

За съжаление, неговата сложност по време е $\Theta(n^2m^2)$. По-бърз алгоритъм се конструира въз основа на следния факт.

Лема 1 *$M[1\,..\,n, 1\,..\,m]$ е масив на Монж тогава и само тогава, когато*

$$M[p, q] + M[p + 1, q + 1] \leq M[p + 1, q] + M[p, q + 1]$$

за $1 \leq p < n$ и $1 \leq q < m$.

Доказателство: В едната посока доказателството е тривиално. Нека

$$X = \{\big(\,((p, q),(s, t)),((p, t),(s, q))\,\big) \mid 1 \leq p < s \leq n, 1 \leq q < t \leq m\}$$
$$Y = \{\big(\,((p, q),(p + 1, q + 1)),((p, q + 1),(p + 1, q))\,\big) \mid 1 \leq p < n, 1 \leq q < m\}$$

Очевидно $Y \subseteq X$. Тогава

$$M[p, q] + M[s, t] \leq M[p, t] + M[s, q], \text{ за } 1 \leq p < s \leq n \text{ и } 1 \leq q < t \leq m$$

влече

$$M[p, q] + M[p + 1, q + 1] \leq M[p, q + 1] + M[p + 1, q], \text{ за } 1 \leq p < n \text{ и } 1 \leq q < m$$

В обратната посока доказателството е по-триково. Допускаме, че

$$M[p, q] + M[p + 1, q + 1] \leq M[p, q + 1] + M[p + 1, q], \text{ за } 1 \leq p < n \text{ и } 1 \leq q < m \quad (1)$$

Ще покажем, че

$$M[p, q] + M[s, t] \leq M[p, t] + M[s, q], \text{ за } 1 \leq p < s \leq n \text{ и } 1 \leq q < t \leq m \quad (2)$$

Първо фиксираме броя на редовете така: $n = 2$. Ще докажем, че

$$M[1, q] + M[2, q + 1] \leq M[2, q] + M[1, q + 1], \text{ за } 1 \leq q < m \quad (3)$$

влече

$$M[1, q] + M[2, t] \leq M[2, q] + M[1, t], \text{ за } 1 \leq q < t \leq m \quad (4)$$

Това ще направим по индукция по m. Базата е $m = 2$. Тогава (3) и (4) стават едно и също неравенство, а именно

$$M[1, 1] + M[2, 2] \leq M[2, 1] + M[1, 2] \quad (5)$$

Базата е доказана. Да допуснем, че за някое $m \geq 2$ е изпълнено, че за **всяка** матрица A с два реда и m колони:

$$\big(A[1,q] + A[2,q+1] \leq A[2,q] + A[1,q+1], \text{ за } 1 \leq q < m\big) \;\rightarrow$$
$$\big(A[1,q] + A[2,t] \leq A[2,q] + A[1,t], \text{ за } 1 \leq q < t \leq m\big) \tag{6}$$

Разглеждаме произволна матрица M с два реда и $m+1$ колони, за която е изпълнено

$$\begin{aligned}
&M[1,1] + M[2,2] \leq M[2,1] + M[1,2]\\
&M[1,2] + M[2,3] \leq M[2,2] + M[1,3]\\
&M[1,3] + M[2,4] \leq M[2,3] + M[1,4]\\
&\dots\\
&M[1,m-1] + M[2,m] \leq M[2,m-1] + M[1,m]\\
&M[1,m] + M[2,m+1] \leq M[2,m] + M[1,m+1]
\end{aligned} \tag{7}$$

Прилагаме индуктивното предположение за подматрицата на M от първите m колони и получаваме

$$M[1,q] + M[2,t] \leq M[2,q] + M[1,t], \text{ за } 1 \leq q < t \leq m \tag{8}$$

Прилагаме индуктивното предположение за подматрицата на M от последните m колони и получаваме

$$M[1,q] + M[2,t] \leq M[2,q] + M[1,t], \text{ за } 2 \leq q < t \leq m+1 \tag{9}$$

Забележете, че всяко неравенствата в (8) или (9) касае две колони в M. Примерно, $M[1,1] + M[2,2] \leq M[2,1] + M[1,2]$ в (8) касае колони 1 и 2. Ето кои двойки колони са “покрити” от (8):

$$\begin{array}{llllll}
(1,2), & (1,3), & (1,4), & \dots, & (1,m-1), & (1,m)\\
 & (2,3), & (2,4), & \dots, & (2,m-1), & (2,m)\\
 & & (3,4), & \dots, & (3,m-1), & (3,m)\\
\dots & & & & & \\
 & & & & (m-2,m-1), & (m-2,m)\\
 & & & & & (m-1,m)
\end{array}$$

Ето кои двойки колони са “покрити” от (9):

$$\begin{array}{llllll}
(2,3), & (2,4), & (2,5), & \dots, & (2,m), & (2,m+1)\\
 & (3,4), & (3,5), & \dots, & (3,m), & (3,m+1)\\
 & & (4,5), & \dots, & (4,m), & (4,m+1)\\
\dots & & & & & \\
 & & & & (m-1,m), & (m-1,m+1)\\
 & & & & & (m,m+1)
\end{array}$$

По отношение на доказателството в индуктивната стъпка, единствената двойка колони, която остава “непокрита” от (8) или (9), е $(1, m+1)$. С други думи, съвкупността на (8) и (9) не ни дава $M[1,1] + M[2,m+1] \leq M[2,1] + M[1,m+1]$. Сега ще изведем и него. Имаме

$$M[1,m] + M[2,m+1] \leq M[2,m] + M[1,m+1] \quad // \text{ от } (7) \tag{10}$$

$$M[1,1] + M[2,m] \leq M[2,1] + M[1,m] \quad // \text{ от } (8) \text{ при } q = 1,\ t = m \tag{11}$$

Сумираме (7) и (8) и получаваме

$$M[1,1] + M[2,m+1] \leq M[2,1] + M[1,m+1] \tag{12}$$

От (8), (9) и (12) заключаваме, че

$$M[1,q] + M[2,t] \leq M[2,q] + M[1,t], \text{ за } 1 \leq q < t \leq m+1$$

Това е краят на доказателството за $n = 2$.

Да се върнем на основното твърдение. Фиксираме произволно $m \geq 2$. Ще докажем импликацията $\boxed{(1) \to (2)}$ по индукция по n. По-подробно, ще докажем, че

$$\begin{aligned}&\big(\, M[p,q] + M[p+1,q+1] \leq M[p,q+1] + M[p+1,q], \text{ за } 1 \leq p < n \text{ и } 1 \leq q < m \,\big) \to \\ &\big(\, M[p,q] + M[s,t] \leq M[p,t] + M[s,q], \text{ за } 1 \leq p < s \leq n \text{ и } 1 \leq q < t \leq m \,\big)\end{aligned} \tag{13}$$

за всяко $n \geq 2$, за същото m, което фиксирахме.

Базата е $n = 2$. Но ние вече доказахме (13) за $n = 2$. Сега да допуснем, че за всяка матрица A с размери $n \times m$, за същото m, което вече фиксирахме, е изпълнено

$$\begin{aligned}&\big(\, M[p,q] + M[p+1,q+1] \leq M[p,q+1] + M[p+1,q], \text{ за } 1 \leq p < n \text{ и } 1 \leq q < m \,\big) \to \\ &\big(\, M[p,q] + M[s,t] \leq M[p,t] + M[s,q], \text{ за } 1 \leq p < s \leq n \text{ и } 1 \leq q < t \leq m \,\big)\end{aligned} \tag{14}$$

Разглеждаме произволна матрица M с размери $(n+1) \times m$, такава че

$$M[p,q] + M[p+1,q+1] \leq M[p,q+1] + M[p+1,q], \text{ за } 1 \leq p < n+1 \text{ и } 1 \leq q < m$$

Ще докажем, че

$$M[p,q] + M[s,t] \leq M[p,t] + M[s,q], \text{ за } 1 \leq p < s \leq n+1 \text{ и } 1 \leq q < t \leq m \tag{15}$$

Нека M' е подматрицата на M от първите n реда. Нека M'' е подматрицата на M от последните n реда. Индуктивното предположение е в сила за M', така че

$$M[p,q] + M[s,t] \leq M[p,t] + M[s,q], \text{ за } 1 \leq p < s \leq n \text{ и } 1 \leq q < t \leq m \tag{16}$$

Индуктивното предположение е в сила и за M'', така че

$$M[p,q] + M[s,t] \leq M[p,t] + M[s,q], \text{ за } 2 \leq p < s \leq n+1 \text{ и } 1 \leq q < t \leq m \tag{17}$$

За да довършим доказателството в индуктивната стъпка, остава да покажем, че

$$M[1,q] + M[n+1,t] \leq M[1,t] + M[n+1,q], \text{ за } 1 \leq q < t \leq m \tag{18}$$

Разглеждаме произволни q и t, такива че $1 \leq q < t \leq m$. От индуктивното предположение знаем, че

$$M[1,q] + M[n,t] \leq M[1,t] + M[n,q] \tag{19}$$

понеже тези четири елемента се намират в M'. От индуктивното предположение също така знаем, че

$$M[n,q] + M[n+1,t] \leq M[n,t] + M[n+1,q] \tag{20}$$

понеже тези четири елемента се намират в M''. Събирайки (19) и (20), получаваме именно (18).

Съвкупността от (16), (17) и (18) е точно (15). Това е и края на доказателството на лемата. □

Използвайки Лема 1, можем с лекота да верифицираме следния алгоритъм, който проверява дали даден масив е масив на Монж.

```
Test Mongeness 2(M[1 .. n, 1 .. m])
1  for p ← 1 to n − 1
2      for q ← 1 to m − 1
3          if M[p,q] + M[p+1,q+1] > M[p,q+1] + M[p+1,q]
4              return 0
5  return 1
```

Коректността следва (почти) директно от лемата. Инвариант за външния цикъл (какъвто не се иска) може да е: при всяко достигане на ред 1, $M[1\,..\,p, 1\,..\,m]$ е масив на Монж. Забележете, че това е истина при първото достигане: тогава $p = 1$, въпросният масив има само един ред и е масив на Монж в празния смисъл (vacuously), понеже неравенствата, дефиниращи масив на Монж (според Лема 1, но и според "каноничната" дефиниция), са такива, че променливите вземат стойности от празното множество, ако коя да е от дименсиите на матрицата е единица.

Сложността по време е очевидно $\Theta(nm)$. Това е линеен алгоритъм в текущия контекст, понеже големината на входа е nm.

Да си припомним, че по условие $\phi(i)$ е номерът на колоната, съдържаща най-левия минимален елемент в ред i на масив на Монж $M[1\,..\,n, 1\,..\,m]$. Нека M е масив на Монж. Иска се да се докаже, че

$$\phi(1) \leq \phi(2) \leq \cdots \leq \phi(n) \tag{21}$$

Твърдението е вярно за примера от условието:

15	19	8	4	25
16	18	5	1	20
50	35	22	10	24
91	64	21	9	5
90	60	16	4	0
100	70	20	4	0
120	75	22	5	1

Да докажем (21) за общия случай. Да допуснем противното: съществува масив на Монж $M[1..n, 1..m]$, такъв че за някое $p \in \{1, \ldots, n-1\}$, най-левият минимален елемент на ред p е в колона t, най-левият минимален елемент на ред $p+1$ е в колона q, и $q < t$. Тогава очевидно

$$M[p,t] + M[p+1,q] = \min \{M[p,j] + M[p+1,k] \mid 1 \leq j, k \leq m\} \tag{22}$$

Да разгледаме елементите $M[p,q]$ и $M[p+1,t]$. По дефиниция на масив на Монж, трябва да е изпълнено

$$M[p,q] + M[p+1,t] \leq M[p+1,q] + M[p,t]$$

Но сумата $M[p,q] + M[p+1,t]$ е една от сумите в дясната страна на (22). Тогава трябва да е изпълнено

$$M[p,t] + M[p+1,q] = M[p,q] + M[p+1,t]$$

Да допуснем, че $M[p,t] < M[p,q]$. Веднага следва, че $M[p+1,q] > M[p+1,t]$. Последното обаче е невъзможно, понеже $M[p+1,q]$ е минимален елемент в ред $p+1$. Тогава $M[p,t] \geq M[p,q]$. Но $M[p,t]$ е минимален елемент в ред p. Тогава $M[p,q] = M[p,t]$. Освен това, $q < t$ по конструкция. Тогава не е вярно, че $M[p,t]$ е най-левият минимален елемент в ред p, противно на първоначалното допускане. Полученото противоречие показва, че първоначалното допускане е грешно. С което твърдението е доказано.

Вече знаем, че $1 \leq \phi(1) \leq \phi(2) \leq \cdots \leq \phi(n) \leq m$. Допускаме, че са дадени

$$\phi(2),\ \phi(4),\ \phi(6),\ \ldots,\ \phi\left(2\left\lfloor\frac{n}{2}\right\rfloor\right)$$

Искаме да намерим

$$\phi(1),\ \phi(3),\ \phi(5),\ \ldots,\ \phi\left(2\left\lceil\frac{n}{2}\right\rceil - 1\right)$$

Предвид $1 \leq \phi(1) \leq \phi(2) \leq \cdots \leq \phi(n) \leq m$, очевидно $\phi(2) \leq \phi(4) \leq \cdots \leq \phi\left(2\left\lfloor\frac{n}{2}\right\rfloor\right)$, а също така

ако n е четно:

$$\begin{aligned} 1 &\leq \phi(1) &&\leq \phi(2) \\ \phi(2) &\leq \phi(3) &&\leq \phi(4) \\ \phi(4) &\leq \phi(5) &&\leq \phi(6) \\ &\cdots &&\cdots \\ \phi(n-2) &\leq \phi(n-1) &&\leq \phi(n) \end{aligned}$$

ако n е нечетно:

$$\begin{aligned} 1 &\leq \phi(1) \leq \phi(2) \\ \phi(2) &\leq \phi(3) \leq \phi(4) \\ \phi(4) &\leq \phi(5) \leq \phi(6) \\ &\cdots \quad \cdots \\ \phi(n-1) &\leq \phi(n) \leq n \end{aligned}$$

Следният алгоритъм решава задачата. Нека функцията ϕ е реализирана чрез масив $\phi[1..n]$. Дадени са стойностите на ϕ на четните позиции. Алгоритъмът попълва останалите позиции. Нека $\mathrm{findmin}(M[p,q], M[p,t])$ е функция, която намира

$$\min \{M[p,q], M[p,q+1], \ldots, M[p,t]\}$$

където $p \in \{1, \ldots, n\}$ и $1 \leq q \leq t \leq m$.

Compute Odd($M[1 \mathinner{..} n, 1 \mathinner{..} m]$, $\phi[1 \mathinner{..} n]$)
  (∗ Дадени са $\phi[2]$, $\phi[4]$, …, $\phi\left[2\left\lfloor\frac{n}{2}\right\rfloor\right]$ ∗)
  $\phi[1] \leftarrow$ findmin$(M[1,1], M[1,\phi[2]])$
  for $k \leftarrow 1$ **to** $\left\lfloor\frac{n}{2}\right\rfloor - 1$
    $\phi[2k+1] \leftarrow$ findmin$(M[2k+1, \phi[2k]], M[2k+1, \phi[2k+2]])$
  if isodd(n)
    $\phi[n] \leftarrow$ findmin$(M[n, \phi[n-1], M[n,m])$

Коректността е очевидна предвид неравенствата, които изведохме горе. Функцията findmin може да се реализира тривиално като търсене в несортиран масив, поради което findmin$(M[p,q], M[p,t])$ работи във време $O(t-q)$ (и използва точно $t-q$ сравнения). Сумарно по всички извиквания на findmin, нейната сложност е $O(m)$; всички изпълнения на findmin използват (общо) точно $m-1$ сравнения. Но "$O(m)$" не описва цялата сложност, защото **for**-цикълът се изпълнява $\Theta(n)$ пъти. Цялата сложност на алгоритъма е $O(n+m)$. Това е точната асимптотична горна граница.

$O(nm)$ също е валидна асимптотична горна граница, но в условието се иска да се покаже $O(n+m)$.

И накрая конструираме алгоритъм, намиращ всички минимуми на редове в масив на Монж. Входът е масив на Монж $M[1 \mathinner{..} n, 1 \mathinner{..} m]$. Генерираме подмасивите M' и M'' съответно от четните и нечетните редове. Правим рекурсивно викане върху M'. След като приключи неговото изпълнение, имаме $\phi[p]$ за четните p и оттам намираме $\phi[p]$ за нечетните p във време $O(n+m)$ по начин, който вече видяхме.

Рекурсията има спирачка $n = 1$: ако подмасивът, върху който изпълняваме алгоритъма, има само един ред, да кажем ред номер p, просто намираме с последователно търсене номера на колоната (между 1 и m), съдържаща най-левия минимум, и записваме този номер във $\phi[p]$. Забележете, че дъното на рекурсията не се изпълнява в константно време! Алгоритмите по схемата **Разделяй-и-Владей** от лекции бяха по правило върху едномерни масиви и дъното на рекурсията беше изпълнение върху масиви с големина $\Theta(1)$, което отнема време $\Theta(1)$. Тук обаче масивът е двумерен, а в рекурсивните викания едната димензия—а именно m—се запазва, така че дъното на рекурсията има сложност по време $\Theta(m)$.

Нека $T(n,m)$ е сложността по време на нашия алгоритъм. В сила е

$$T(n,m) = T(n/2, m) + n + m$$

Събираемото $T(n/2, m)$ отразява факта, че се прави точно едно рекурсивно викане върху масив с наполовина по-малко редове, но същият брой колони. Събираемото $n+m$ отразява факта, че извън рекурсивното викане се върши работа, която е линейна в $n+m$. Но ние няма да използваме това равенство, защото не се иска решение с Θ.

Тъй като се иска само (асимптотична) горна граница за сложността по време, достатъчно е да разгледаме неравенството

$$T(n,m) \leq T(n/2, m) + cn + dm$$

Няма да доказваме решението строго по индукция, това не се иска. С развиване имаме

$$\begin{aligned}
T(n,m) &\leq T(n/2,m) + cn + dm \\
&\leq (T(n/4,m) + cn/2 + dm) + cn + dm \\
&= T(n/4,m) + cn(1/2+1) + 2dm \\
&\leq (T(n/8,m) + cn/4 + dm) + c(n/2+n) + 2dm \\
&= T(n/8,m) + cn(1/4+1/2+1) + 3dm \\
&\dots \\
&\leq T\left(\frac{n}{2^i}, m\right) + cn \underbrace{\left(\frac{1}{2^{i-1}} + \frac{1}{2^{i-2}} + \dots + \frac{1}{2^1} + \frac{1}{2^0}\right)}_{\text{ограничено от константа за всяко } i} + idm
\end{aligned}$$

Максималното i е $i_{\max} = \Theta(\lg n)$. Нека $2^{i_{\max}} = a$ за някаква константа a. Замествайки i с $i_{\max}$ и имайки предвид, че сумата на геометричната прогресия е ограничена от константа, имаме право да кажем, че

$$T(n,m) \leq T(a,m) + c_1 n + d_1 m \lg n$$

за някакви константи c_1 и d_1. Както вече отбелязахме, $T(a,m)$ има сложност по време $\Theta(m)$, което се "поглъща" в $d_1 m \lg n$. Имаме право да кажем, че

$$T(n,m) = O(n + m \lg n)$$