

вариант	ф. номер	група	поток	курс	специалност
1					
Име:					

Контролно по Бързи алгоритми върху структури от данни
спец.

06.12.2018 г.

Задача 1. Разглеждаме следния проблем:

Дадено: $A[0..n-1][0..m-1]$ - матрица от различни числа.

Вход: $0 \leq i \leq j < n$, $0 \leq k \leq l < m$.

Изход: $(s, r) \in [i; j] \times [k; l]$, за която $A[s][r] = \min A[i..j][k..l]$.

1. (0.5 т.) Предложете алгоритъм, който индексира дадената матрица за време $O(nm \log n \log m)$ и отговаря на всяка заявка за време $O(1)$.
2. (0.5 т.) Предложете индекс, който зависи от $O(nm)$ сравнения върху елементите на дадената матрица и има характеристиките от първата подточка. Докажете, че сравненията необходими за построяването на индекса действително са $O(nm)$.
3. (0.5 т.) Предложете алгоритъм, който индексира дадената матрица за време $O(nm)$ и отговаря на всяка заявка за време $O(1)$.

Обосновете коректността и времевата сложност на предложените от Вас алгоритми.

Упътване: Може да си мислите за подматриците $A[s2^p..(s+1)2^k-1][t2^q..(t+1)2^l-1]$ на A , организирани по степените на двойката $2^p < n$ и $2^q < m$. Показателен е случаят, когато m и n са точни степени на 2.

Забележка: Алгоритмите, разглеждани по време на курса могат да използват без допълнителна верификация.

вариант	ф. номер	група	поток	курс	специалност
1					
Име:					

Контролно по Бързи алгоритми върху структури от данни
спец.

06.12.2018 г.

Задача 1. Разглеждаме следния проблем:

Дадено: $A[0..n-1][0..m-1]$ - матрица от различни числа.

Вход: $0 \leq i \leq j < n$, $0 \leq k \leq l < m$.

Изход: $(s, r) \in [i; j] \times [k; l]$, за която $A[s][r] = \min A[i..j][k..l]$.

1. (0.5 т.) Предложете алгоритъм, който индексира дадената матрица за време $O(nm \log n \log m)$ и отговаря на всяка заявка за време $O(1)$.
2. (0.5 т.) Предложете индекс, който зависи от $O(nm)$ сравнения върху елементите на дадената матрица и има характеристиките от първата подточка. Докажете, че сравненията необходими за построяването на индекса действително са $O(nm)$.
3. (0.5 т.) Предложете алгоритъм, който индексира дадената матрица за време $O(nm)$ и отговаря на всяка заявка за време $O(1)$.

Обосновете коректността и времевата сложност на предложените от Вас алгоритми.

Упътване: Може да си мислите за подматриците $A[s2^p..(s+1)2^k-1][t2^q..(t+1)2^l-1]$ на A , организирани по степените на двойката $2^p < n$ и $2^q < m$. Показателен е случаят, когато m и n са точни степени на 2.

Забележка: Алгоритмите, разглеждани по време на курса могат да използват без допълнителна верификация.