

АиСП - фебруар 2020. - практични део испита

Упутство

Преименовати тренутни фолдер у `mixхуу_Ime_Prezime_AA` где `xx` представља последње две цифре године уписа факултета, `ууу` представља број индекса и `AA` представља иницијале асистента (`ID` или `SS`). Нпр. `mi18999_Petar_Petrovic_ID`.

Све задатке радити искључиво у приложеним `.cpp` датотекама. Остале `.cpp` датотеке неће бити прегледане.

Најбитнија карактеристика алгоритама је коректност и она се најстрожије бодује. Решења која су коректна, али не испуњавају тражене сложености, добијају одређен проценат бодова зависно од сложености.

На располагању Вам је скрипта помоћу које аутоматски можете тестирати своја решења. Покреће се позиционирањем у овај фолдер у терминалу и позивањем команде `./tester.sh` након чега ће Вам бити приказане даље инструкције.

Прасе и вук [10 поена]

Једно од три прасета је одлучило да направи кућу од штапова како би се заштитило од вука. Битно је да кућа буде издржљива и да има основу облика правоуглог троугла. Прасе жели да зна на колико различитих начина може, од штапова које има на располагању, да направи правоугли троугао (ако употреби тачно један штап за сваку страну). Написати програм који реализује алгоритам који за задате дужине штапова одређује број правоуглих троуглова који се могу конструисати. Временска сложеност алгоритма треба да буде $O(n^2)$, а просторна сложеност $O(n)$.

Улаз: Са стандардног улаза се учитава број штапова n ($1 \leq n \leq 5000$). Затим се учитавају дужине штапова, тј. n различитих целих бројева из интервала $[1, 20000]$.

Излаз: На стандардни излаз исписати број правоуглих троуглова који се могу конструисати од задатих штапова.

Пример 1

Улаз:

6
56 16 65 33 63 10

Излаз:

2

Објашњење: Могуће је направити правоугле троуглове чије су странице (33, 56, 65) и (16, 63, 65).

Пример 2

Улаз:

4
5 2 4 3

Излаз:

1

Објашњење: Могуће је направити правоуги троугао чије су странице (3, 4, 5).

Скокови [10 поена]

Дато је n поља поређаних у низ. На сваком пољу је уписан по један цео број. Играч се на почетку налази на првом пољу и треба да дође до последњег. Играч може да се помери на следеће поље, или да прескочи следеће поље и стане на поље иза тог. Када дође до последњег поља, играч сабира бројеве са свих поља преко којих је прешао. Написати програм који имплементира алгоритам за одређивање највећег збира који играч може да оствари на тај начин. Временска сложеност алгоритма треба да буде $O(n)$, а просторна сложеност $O(1)$.

Улаз: Са стандардног улаза се учитава број поља n ($1 \leq n \leq 10^6$), након чега се учитава n целих бројева из интервала $[-10^9, 10^9]$. Збир апсолутних вредности свих учитаних бројева је највише 10^9 .

Излаз: На стандардни излаз исписати највећи збир који је могуће остварити на путу од првог до последњег поља на описани начин.

Пример 1

Улаз:

5
-1 2 -3 -4 5

Излаз:

3

Објашњење: Највећи збир се остварује сабирањем бројева -1, 2, -3 и 5.

Пример 2

Улаз:

5
1 -2 -3 -4 5

Излаз:

3

Објашњење: Највећи збир се остварује сабирањем бројева 1, -3 и 5.

Опадајући низ [10 поена]

Низ бројева је опадајући уколико је сваки његов елемент мањи или једнак претходном. На улазу се задаје опадајући низ бројева. Написати програм који реализује алгоритам за одређивање наредног опадајућег низа (у лексикографском поретку већег), при чему су сви елементи низа из скупа $1, 2, \dots, n$. Временска и просторна сложеност алгоритма треба да буду $O(n)$.

Улаз: Са стандардног улаза се учитава дужина низа n након чега се учитава опадајући низ од n ($1 \leq n \leq 10^6$) целих бројева из интервала $[1, n]$.

Излаз: На стандардни излаз исписати лексикографски следећи опадајући низ уколико постоји, или **нема** уколико не постоји.

Пример 1

Улаз:

5
5 4 4 2 2

Излаз:

5 4 4 3 1

Пример 2

Улаз:

5
5 5 5 5 5

Излаз:

нема